

Presentación

Nos complace presentar una edición más de las memorias del premio de Instrumentación y Electrónica “Gustavo del Castillo y Gama” en su versión 2018. En esta ocasión se han inscrito 13 proyectos desarrollados en dependencias académicas de la Universidad tanto de los campus de San Luis Potosí como del campus de Salinas de Hidalgo y de Instituciones académicas de la Ciudad. Los temas de los proyectos son diversos, desde el control de robots, tanto terrestres como aéreos, el desarrollo de algoritmos computacionales para el procesamiento y análisis de datos en particular para las comunicaciones, sistemas de espectroscopia así como de sistemas que involucran instrumentación optoelectrónica. Los proyectos presentados en esta versión del premio, han sido fundamentalmente originados en proyectos de licenciatura y/o posgrado desarrollados en buen porcentaje en las instalaciones del Instituto. Estos nos llenan de gran satisfacción y nos hace pensar en el excelente trabajo de docencia e investigación que se ha venido desarrollando durante los últimos veinticinco años en nuestro instituto. Contamos con participantes de la Facultad de Ciencias, de la Facultad de Ingeniería, de la Coordinación Académica Región Altiplano Oeste (Salinas de Hidalgo), de la Universidad Tangamanga y del Instituto de Investigación en Comunicación Óptica. Agradecemos a los jurados del premio por el interés y profesionalismo que mostraron y la disposición de su tiempo para la revisión minuciosa de los extensos de los proyectos participantes. En esta ocasión el jurado estuvo formado por los Profesores Mtro. Jorge Rodríguez Rodríguez de la Facultad de Ingeniería, Dr. Hernán González Aguilar de la Facultad de Ciencias y del Dr. José Manuel Flores Camacho del Instituto de Investigación en Comunicación Óptica. Muchas gracias nuevamente.

Comité organizador

San Luis Potosí, S.L.P. Mayo 2018

Rector

Arq. Manuel Fermín Villar Rubio

Secretario General

Dr. en Arq. Anuar Abraham Kasis Ariceaga

Director

Dr. Alfonso Lastras Martínez

Comité organizador del concurso

Dr. Raúl Eduardo Balderas Navarro

Ing. Jorge Antonio González Fortuna

Dr. Luís Felipe Lastras Martínez

Dra. Marcela Mejía Carlos

Ing. Liliana del Carmen Mendoza Tello

Dr. Oscar F. Núñez Olvera

Maestro Esequiel Ontiveros Hernández

Ing. Francisco Ramírez Jacobo

M. en I. Griselda Rodríguez Pedroza

Jurado

Dr. José Manuel Flores Camacho

Dr. Hernán González Aguilar

Mtro. Jorge Rodríguez Rodríguez

Edición de memorias

Dra. Marcela Mejía Carlos

Dr. Luis Felipe Lastras Martínez

Ing. Liliana del Carmen Mendoza Tello

Índice

		Pág.
1	Algoritmo cifrado de Imágenes paralelizable basado en sincronización de autómatas celulares, cajas de sustitución y mapas caóticos Espíndola Paizano Blanca Jazmin, Ramírez Torres Marco Tulio	1
2	Caracterización Empírica de Canales de Comunicaciones Vehiculares en 700 MHz C. A. Gómez Vega, C. A. Gutiérrez	5
3	I. Prototipo de espectrómetro compacto ajustable Esdras Terranova Azuara Juárez	9
4	Implementación de sistemas bioinspirados L. F. De La Rosa García, L. J. Ontañón-García	13
5	Filtro DB L. F. De La Rosa García, D. M Quiroz Moreira, B.M Rodriguez Ovalle, A. García Paisano	17
6	Diseño y construcción de una máquina de control numérico D. Gallegos Baez, F. Sánchez Niño	19
7	Diseño y prototipado de humanoide por impresión 3D Israel Macias González, Juan Almendárez Rodríguez	23
8	Luminaria LED auto recargable sin necesidad de paneles solares Israel Macias González, Juan Almendárez Rodríguez	27
9	Robot Autónomo-Teleoperado Juan Alberto Vértiz Hernández, Lorena Rodríguez Salas, Omar Vital Ochoa	31
10	Control Servo Visual mediante OCR y Seguimiento de objetivos de un QuadRotor AR.Drone en LabVIEW Juan Manuel Fortuna Cervantes, Marco Tulio Ramirez Torres	37
11	Compresión de señales usando transformadas ortogonales Reyes-López L.E., Castro García R	43

12	Robot movil con una trayectoria basada en un sistema caotico de lorenz	47
	Domínguez-Rivera I., Villa Salas L.R., L.J. Ontañón García	
13	Rutinas específicas programadas con máquinas de estados en LabVIEW para AR.DRONE 2.0	51
	Martínez López Maximiliano, Ramírez Torres Marco Tulio	

Algoritmo cifrado de Imágenes paralelizable basado en sincronización de autómatas celulares, cajas de sustitución y mapas caóticos

Espíndola Paizano Blanca Jazmin¹, Ramírez Torres Marco Tulio¹.

¹Coordinación Académica Región Altiplano Oeste, UASLP, Salinas, S.L.P. blancaep_96@hotmail.com

Resumen—En esta investigación se presenta una nueva versión de cifrado de imágenes en forma paralela, en conjunto con su validación, con el fin de obtener un algoritmo de rápida operación sin descuidar aspectos de seguridad criptográfica y/o perceptual. El algoritmo utiliza dos procedimientos: sustitución basada en la sincronización de autómatas celulares y las denominadas S-box o cajas de sustitución, mientras que la operación de difusión es a través de la técnica de mapas caóticos llamada Arnold's cat map. Para solventar los problemas de alta correlación adyacente que presentan las imágenes, se utilizan datos externos al coeficiente del pixel como lo son sus coordenadas renglón y columna. Utilizando las cajas de sustitución o S-box es posible cancelar la correlación entre las posiciones de cada uno de los pixeles permitiendo procesar los datos con la sustitución basada en autómatas celulares. Posteriormente los datos pasan por la técnica de Arnold's cat map. Cabe destacar que todos los procesos funcionan de manera paralela omitiendo así la retroalimentación o encadenamiento.

Por lo que este algoritmo podría ser una opción viable para cifrar grandes cantidades de datos redundantes como los existentes en las imágenes digitales, e incluso junto con técnicas de supercómputo podría ser una opción segura para cifrar video en tiempo real.

I. INTRODUCCIÓN

A lo largo de los años ha habido una importante expansión tecnológica, ahora se pueden realizar diferentes operaciones a través de internet y, con esto, es necesario proteger nuestros datos personales. Gran parte de nuestra información se encuentra en imágenes digitales como diagramas, imágenes médicas, imágenes personales, etc., y esta información se almacena en dispositivos o se envía a través de internet. Para proteger la confidencialidad de estos datos, se han propuesto varias técnicas con diferentes enfoques. Por ejemplo, encriptación, esteganografía, marca de agua, entre otros. El cifrado de imágenes es un área de investigación especial, ya que requiere conocimientos y consideraciones específicas ya que las imágenes tienen alta redundancia, grandes volúmenes, operaciones en tiempo real y datos comprimidos de un determinado formato [1]. El objetivo de esta técnica es transformar la información en una forma ininteligible. Por lo tanto, los algoritmos de encriptación de imágenes deberían proporcionar dos tipos de seguridad: criptográfica y perceptual [1]. Esta técnica podría ser una opción para proteger la

información cuando se almacena o cuando se envía a través de una red, solo las personas autorizadas pueden tener acceso a la información en forma perceptible. En este trabajo se presenta un algoritmo de confusión y difusión paralelizable para el cifrado de imágenes basado en la sincronización de autómatas celulares y sustitución con S-Box para el proceso de confusión, y el mapa de Arnold para el proceso de difusión.

II. METODOLOGIA

La implementación numérica de este sistema de cifrado se llevó a cabo bajo el lenguaje de programación gráfico LabVIEW, donde cada función se programa en un bloque.

Para el algoritmo de criptografía se han usado imágenes ampliamente utilizadas en el procesamiento de imágenes en escala de grises como: Lena, mandril y pimientos.

Se utilizó como base el sistema de Cifrado basado en la Sincronización de Autómatas Celulares (CSAC), éste contiene un generador de números pseudoaleatorios (PRNG). La función principal se denomina h y requiere de dos vectores x e y de n bits y $n + 1$ bits respectivamente. Para calcular una secuencia pseudoaleatoria, la función requiere que el autómata celular retroceda en el tiempo siguiendo el orden de la regla 90 [2].

El PRNG incluye una retroalimentación para generar diferentes números conforme evoluciona en el tiempo. Para nuestro propósito, no consideramos el proceso de retroalimentación y usamos un píxel de la imagen llamado m en lugar del vector inicial x , con un condición inicial z en lugar de y . El vector resultante se llama m' .

El sistema de cifrado utiliza cuatro vectores como condiciones iniciales: $z0$, $z1$, $z2$ y $z3$. El algoritmo comienza al calcular el número de iteraciones para el mapeo caótico. Este número se calcula con la ecuación (1)

$$\text{Iteraciones} = sm \oplus z0 \oplus m2 \quad (1)$$

Donde sm es el coeficiente del primer píxel de la imagen sustituido en el S-Box. $z0$ es una condición inicial y $m2$ representa el segundo coeficiente diferente a $m1$. Este

iguales a cero. Las imágenes están cifradas bajo las mismas condiciones iniciales (misma clave). La versión cifrada de la imagen sólida se llama máscara. Y, por último, tratamos de recuperar la imagen del panda mediante una operación XOR entre las dos imágenes cifradas pixel por pixel. En la Fig. 3 se muestra este ataque. La imagen simple recuperada se muestra en la figura 3e), y una inspección visual revela la eficacia de nuestro esquema de cifrado propuesto. También el histograma de la imagen recuperada se muestra en la Fig. 4. El resultado es un histograma uniforme.

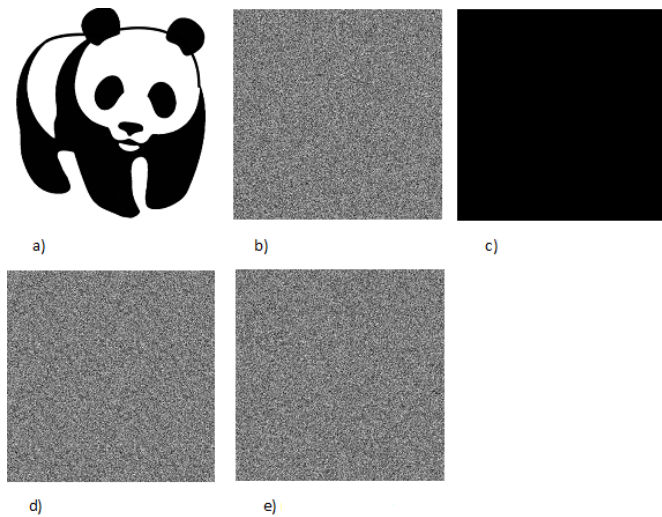


Fig. 3. Chosen plain text-attack para la imagen de prueba panda. (a) La imagen original, (b) la imagen cifrada, (c) la imagen plana escogida, (d) la imagen enmascarada y (e) la imagen recuperada.

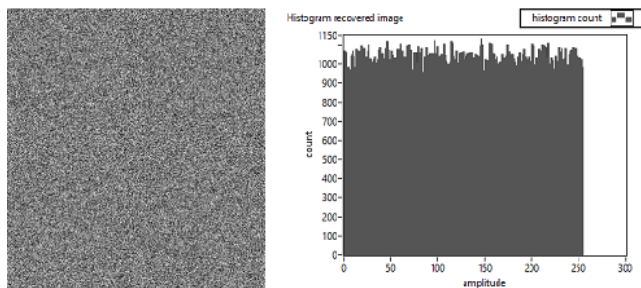


Fig. 4. Histograma de la imagen recuperada

IV. CONCLUSIONES

En este trabajo se propone un sistema de encriptación de imágenes basado en fenómenos de sincronización de AC, sustitución con S-Box y permutación con el mapa de Arnold's cat. Los resultados del análisis de seguridad muestran que este esquema propuesto pasa por diferentes pruebas y ataques de criptoanálisis. La seguridad perceptual no se ve comprometida, incluso cuando el algoritmo funciona en forma paralela.

En un futuro, trabajaremos con pruebas de criptoanálisis diferenciales y finalmente haremos una implementación de hardware.

V. REFERENCIAS

- [1] S. LIAN: Multimedia content encryption: Techniques and applications. 1st edn. Auerbach Publications, 6000 Broken Sound Parkway NW (2009).
- [2] J. URIAS, E. UGALDE, G. SALAZAR: A cryptosystem based on cellular automata. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 8(4). AIP (1998) 819–822.
- [3] A. M. DEL REY, G. R. SÁNCHEZ, & A. DE LA VILLA CUENCA, Encrypting digital images using cellular automata. In *International Conference on Hybrid Artificial Intelligence Systems*. Springer, Berlin, Heidelberg (2012) 78–88.

Caracterización Empírica de Canales de Comunicaciones Vehiculares en 700 MHz

C. A. Gómez Vega, C. A. Gutiérrez
Facultad de Ciencias
Universidad Autónoma de San Luis Potosí
San Luis Potosí, México
Email: cgomez@ieee.org, cagutierrez@ieee.org

Resumen—En este reporte se presenta la implementación de un banco de pruebas para realizar mediciones del espectro de potencia Doppler (EPD) de canales de comunicaciones vehiculares en 700 MHz. Este banco de pruebas es construido con equipo de propósito general, por lo que se puede replicar fácilmente. Para demostrar el valor práctico de la configuración experimental, se llevó a cabo una campaña de medición para recolectar datos de canales de comunicación de vehículo a vehículo (V2V, por sus siglas en inglés) y de vehículo a infraestructura instalada en el camino (V2I, por sus siglas en inglés). Con los datos recolectados se obtuvieron los momentos estadísticos del EPD: el desplazamiento Doppler promedio y la dispersión Doppler. La configuración experimental y los resultados obtenidos sirven como referencia para la caracterización empírica del canal, así como para el diseño y evaluación de esquemas de transmisión para sistemas de comunicación vehicular.

Palabras clave —Comunicaciones vehiculares, efecto Doppler, mediciones de canal, momentos estadísticos, sondeo de canal.

I. INTRODUCCIÓN

Los sistemas de transporte inteligente (STIs) han surgido como una solución tecnológica a los problemas de tráfico y seguridad vial. Estos sistemas integran tecnologías de información y comunicación con aplicaciones para asistir a los conductores en la toma de decisiones. La capacidad de alertar a los vehículos sobre situaciones de riesgo y obtener información sobre las condiciones de tráfico depende de la existencia de enlaces de comunicación de vehículo a vehículo (V2V, por sus siglas en inglés) y de los vehículos a la infraestructura instalada en el camino (V2I, por sus siglas en inglés) [1]. Las características de dichos enlaces han motivado el desarrollo de sistemas de comunicación de corto alcance (DSRC, por sus siglas en inglés) que permiten la transmisión de datos sobre enlaces V2V y V2I de poca distancia (menor a 1 km) [1].

Con el objetivo de apoyar el desarrollo de la tecnología DSRC, en EE. UU. y Europa han sido asignados canales dedicados en la banda de 5.9 GHz. También, el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés) ha publicado el estándar IEEE 802.11p para homologar las características de la capa física de los sistemas DSRC. Además de estos esfuerzos de estandarización, existe un interés en la banda de 700 MHz, ya que esta banda ha sido considerada en Japón para tecnología DSRC. En este país, la

Asociación de Industrias y Empresas de Radiofrecuencia (ARIB, por sus siglas en inglés) ha publicado el estándar STD-T109 para sistemas DSRC en la banda de 700 MHz [1].

Además de seguir las especificaciones de los estándares, el diseño de los esquemas de transmisión para sistemas DSRC debe considerar las características de los canales de propagación V2V y V2I [2]. En entornos vehiculares, el canal se ve afectado por diversos factores, como las condiciones de movilidad de los vehículos que interactúan entre sí, la localización de las antenas, así como la densidad de objetos interferentes presentes en el entorno de propagación. Además, se ha demostrado que el canal vehicular tiene las características de un proceso estocástico no estacionario [2]. Aunque es importante desarrollar modelos teóricos de canal que proporcionen una abstracción matemática de la propagación de señales de radiofrecuencia en entornos vehiculares, es necesario realizar mediciones de campo para obtener información realista de las características del canal que sirva no solamente para validar modelos teóricos, sino también para desarrollar modelos empíricos.

Para realizar mediciones de campo que permitan la caracterización empírica del canal, se requiere desarrollar un sondeador de canal. En el caso de banda ancha, implementar este sistema de medición es una tarea compleja y costosa que requiere equipo de propósito específico cuya adquisición suele estar fuera del presupuesto de la mayoría de las universidades. Una alternativa que permite obtener información parcial de las estadísticas del canal utilizando equipos menos costosos y de propósito más general consiste en utilizar generadores de señales de RF, analizadores vectoriales de redes o analizadores de espectro. Estos equipos se pueden emplear fácilmente para hacer una caracterización empírica de canales V2V y V2I de banda angosta. Aunque los bancos de pruebas de este tipo tienen la capacidad de medir la dispersión de la señal en el dominio de la frecuencia caracterizada por el espectro de potencia Doppler (EPD), por lo general solamente se han utilizado para medir las pérdidas por propagación y la distribución de la envolvente de la señal recibida (desvanecimientos), e. g., ver [3].

El EPD provee información relevante sobre la naturaleza variante en el tiempo de los canales V2V y V2I [4]. Con el objetivo de estudiar estas propiedades, en este proyecto se desarrolló un banco de pruebas para mediciones del EPD de canales de comunicación V2V y V2I. Este banco de pruebas

se implementó utilizando equipo de propósito general que comúnmente está disponible en las universidades: un analizador de espectro y un generador de señales de RF de hasta 800 MHz. Para mostrar el valor práctico de la configuración experimental, se presentan los resultados de una campaña de medición del EPD de canales V2I a 700 MHz. Los datos obtenidos fueron procesados para calcular el primer momento y la raíz cuadrada del segundo momento central del EPD, es decir, la frecuencia Doppler promedio y la dispersión Doppler. El banco de pruebas se puede replicar fácilmente, y los resultados presentados se pueden utilizar como referencia para el diseño y evaluación de esquemas de transmisión para comunicaciones vehiculares en la banda de los 700 MHz.

El resto del reporte está organizado de la siguiente manera: En la Sección II se describe el banco de pruebas para realizar las mediciones. En la Sección III se describe el escenario en el que se llevaron a cabo las mediciones y se presentan los resultados obtenidos para comunicaciones V2I. Finalmente, en la Sección IV se presentan las conclusiones de este trabajo.

II. BANCO DE PRUEBAS

El principio básico de la medición de canal se basa en transmitir una señal conocida que “excita” al canal de forma que en el receptor se observa una señal que se propagó a través del canal inalámbrico y que contiene información sobre la respuesta al impulso del canal o sobre sus autovalores [5]. Al conocer ambas señales, la transmitida y la recibida, se puede estimar el comportamiento del canal mediante diversas técnicas. Uno de los métodos consiste en medir el EPD de canales de banda angosta para observar la dispersión de la señal en el dominio de la frecuencia provocada por los desplazamientos Doppler. Estas mediciones se realizan transmitiendo una señal sinusoidal continua en la frecuencia deseada (autofunción) y midiendo el espectro de potencia de la señal recibida en el dominio de la frecuencia (autovalores). Idealmente, en un enlace directo con terminales estáticas, se debe observar un tono puro en la frecuencia transmitida. Sin embargo, debido a que en un entorno vehicular las terminales se encuentran en movimiento, el tono se distorsionará y su energía se dispersará en el dominio de la frecuencia debido a la propagación multitrayectoria y a los desplazamientos Doppler. De esta forma, el espectro medido ya no mostrará un único pico de señal, sino múltiples picos generados por la dispersión de la señal. Este es el principio que se explota en este trabajo para caracterizar el EPD de canales V2V y V2I.

Para implementar el banco de pruebas se utilizó una configuración sencilla para medir la dispersión en el dominio de la frecuencia bajo el principio descrito empleando equipo de propósito general. En la Fig. 1 se muestra un diagrama a bloques del sistema de medición compuesto por un generador de señales de RF Keysight N9310A y un analizador de espectro Keysight Fieldfox N9913A como transmisor y receptor, respectivamente. En el caso de las mediciones de canales V2V, el generador de RF (estación transmisora) y el analizador de espectro (estación receptora) se montan en diferentes vehículos, mientras que para canales V2I la estación transmisora se encuentra fija en una ubicación conocida. En el setup de la Fig. 1, las computadoras personales (PC) y los GPS permiten guardar tanto la información del EPD como hacer un

registro de la posición instantánea de cada estación. En el lado del transmisor, se utiliza la PC solamente para grabar la posición de la estación, mientras que, en el lado de la estación receptora, la PC captura las mediciones del analizador de espectro y los datos del GPS. Cabe mencionar que para las mediciones de V2I en las que el transmisor está en un punto estático, basta con capturar una sola lectura del GPS.

El generador de señales se configuró para transmitir a una potencia de 20 dBm con una frecuencia portadora de 760 MHz, la cual es la que define el estándar ARIB STD-T109 como frecuencia central. En México, esta frecuencia está reservada para canales de televisión digital. Sin embargo, la banda aún no está en uso, como se pudo corroborar a partir de mediciones. Por su parte, el analizador de espectro se configuró con una resolución adecuada para observar los efectos del canal en el dominio de la frecuencia. Para ello se estableció una tasa de muestreo (tiempo de barrido) lo suficientemente rápida para captar las variaciones de la señal. Se eligió un span de 1 KHz con 601 muestras en el dominio de la frecuencia y ancho de banda de resolución de 6 Hz. Con esta resolución en el dominio de la frecuencia se obtuvo un tiempo de muestreo de 270 ms. Para manejar la potencia de la señal recibida se configuró el analizador de espectro con una atenuación de 0 dB con el preamplificador activado. Además, para la transmisión y recepción de la señal, se diseñaron antenas tipo dipolo de media longitud de onda con acopladores de impedancia de media onda para realizar la conexión entre una línea de transmisión desbalanceada (cable coaxial) con una línea balanceada (antena). Al realizar la caracterización de estas antenas se observó que el acoplamiento de impedancias mejora el desempeño del sistema. Las antenas construidas fueron colocadas sobre el techo de los vehículos cerca del parabrisas frontal. Por otro lado, en el caso de comunicaciones V2I, una de las antenas se coloca en una locación fija cerca de la ruta.

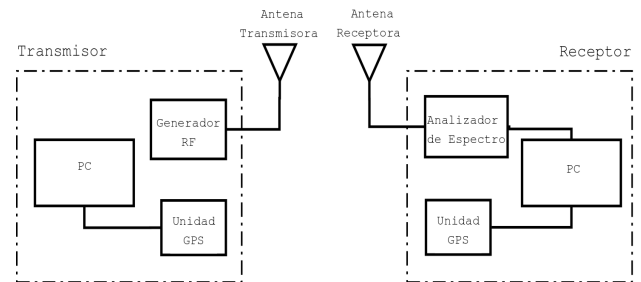


Fig. 1. Diagrama del setup de medición.

III. RESULTADOS ESPECTRALES Y ESTADÍSTICOS

A. Entorno de Mediciones

La campaña de mediciones fue llevada a cabo en la ciudad de San Luis Potosí, México, a las afueras del campus universitario en la Av. Salvador Nava. El entorno de medición es de carácter urbano constando de tres carriles en cada dirección con tráfico vial moderado. En los costados de la vía existen diversas construcciones con alturas variadas y árboles en las banquetas de ambos lados. En el caso de comunicaciones V2I, la antena transmisora se fijó en un costado de la vía a una altura de 3 m. La ruta realizada tiene

una longitud de 1.10 km, y se recorrió con velocidades promedio de 60 km/hr. Durante el trayecto, el vehículo comienza acercándose a la antenna transmisora para posteriormente cruzarla y alejarse de ella. Durante el trayecto se adquirieron 100 capturas del EPD, resultando en un tiempo de medición total aproximado de 70 s debido al tiempo de adquisición de datos en la PC.

B. Mediciones de Canal V2I

Los espectros medidos de los canales de comunicación V2I presentan componentes de línea de vista que aparecen y desaparecen debido a obstrucciones cercanas a la antenna transmisora. En la Fig. 2 se muestra uno de los espectros medidos en vista superior. La frecuencia central del ancho de banda corresponde a 760 MHz y se observan las capturas del EPD medidas durante todo el trayecto. El nivel promedio del ruido observado es de -150 dBm. Se puede notar que la señal recibida se dispersa alrededor de la frecuencia central, concentrándose la energía en una región durante todo el trayecto. Esta región es acotada por el valor de $\pm f_{\text{máx}}$, el cual es el valor del desplazamiento Doppler máximo que puede experimentar la señal cuando el vehículo se mueve con cierta velocidad, que corresponde a 42 Hz considerando la velocidad de 60 km/h. No obstante, la distribución de la energía varía, ya que los picos de potencia observados en cada muestra cambian conforme se avanza en la ruta. Se puede observar que la potencia máxima (puntos de color rojo) del espectro va variando su posición en la frecuencia conforme avanza la ruta. Al inicio del trayecto se observa una región de alta potencia en el sector positivo de la frecuencia debido a que el vehículo va acercándose a la antenna transmisora cuya posición está fija al lado de la calle. Después de esto, se observa que el pico de potencia cambia su ubicación de un valor positivo de frecuencia a un valor negativo. Esto se debe a que durante la ruta el vehículo se acerca a la antenna transmisora y eventualmente pasa frente a ella, para posteriormente irse alejando. Gráficamente puede notarse la naturaleza de un proceso no estacionario al distribuirse la potencia de forma aleatoria sobre la región.

Para estudiar el grado de dispersión causado por el efecto Doppler en los espectros medidos, se calcularon los momentos estadísticos del EPD promedio de todas las mediciones. Para realizar este promedio se procesaron los datos tomando como punto de referencia el instante en que se pasa frente a la antenna. Se consideró una ventana de tiempo corta en que se presenta el cambio al pasar frente a la antenna en el cual las muestras coincidían. Los resultados del análisis de los momentos del EPD del canal V2I se muestran en la Fig. 3. Se puede observar un desplazamiento positivo al iniciar la ruta de aproximadamente 40 Hz, que al acercarse lo suficiente a la antenna disminuye su magnitud hasta alcanzar la frecuencia central en el instante que el vehículo está frente a ella. Posteriormente, la magnitud del desplazamiento aumenta su magnitud con signo negativo al alejarse de la antenna hasta alcanzar un desplazamiento promedio de aproximadamente 35 Hz. Este valor refleja que en la parte final de la ruta la velocidad disminuyó debido al tráfico vehicular durante las mediciones. Por su parte, la dispersión Doppler se mantiene

casi constante durante todo el trayecto. Las variaciones observadas en los resultados mostrados son debido a los cambios de velocidad durante el trayecto.

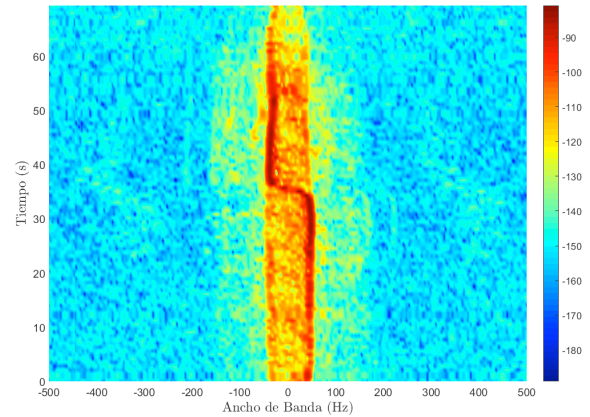


Fig. 2. Medición del EPD de un canal V2I en vista superior.

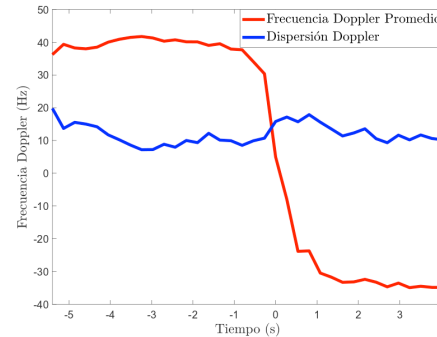


Fig. 3. Momentos estadísticos de los canales V2I medidos.

IV. CONCLUSIONES

En este proyecto se implementó un banco de pruebas para realizar mediciones del EPD de canales V2I y V2V. Esta configuración experimental es flexible ya que puede utilizarse para medir las características de la envolvente, así como la dispersión en la frecuencia debido al efecto Doppler. Con este setup se realizó una campaña de mediciones de canales V2V y V2I cuyos resultados espectrales y estadísticos sirven como referencia para la caracterización empírica del canal.

REFERENCIAS

- [1] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [2] W. Viriyasitavat *et al.*, "Vehicular Communications: Survey and Challenges of Channel and Propagation Models," *IEEE Veh. Technol. Mag.*, vol. 10, no. 2, pp. 55–66, 2015.
- [3] H. Fernández *et al.*, "Path Loss Characterization for Vehicular Communications at 700 MHz and 5.9 GHz Under LOS and NLOS Conditions," *IEEE Antennas and Wirel. Propag. Lett.*, vol. 13, pp. 931–934, 2014.
- [4] C.-X. Wang *et al.*, "Vehicle-to-vehicle channel modeling and measurements: Recent advances and future challenges," *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 96–103, Nov. 2009.
- [5] A. Molisch, *Wireless Communications*. Chichester, UK: John Wiley and Sons, 2005.

Prototipo de espectrómetro compacto ajustable

Esdras Terranova Azuara Juárez
 Instituto de Investigación en Comunicación Óptica
 Universidad Autónoma de San Luis Potosí
 San Luis Potosí, México
 Esdras.Terranova@hotmail.com

Resumen- Se describe el diseño óptico y electrónico de un espectrómetro compacto ajustable, con un rango de medición en las secciones del espectro ultravioleta y visible, una resolución espectral de 3.3 nm y 1348 pixeles. El instrumento controla una cámara lineal cuya señal es digitalizada, a través de un convertidor analógico a digital de 12 bits, la información espectral es enviada por medio de comunicación USB a un ordenador personal, para ser procesada, analizada y visualizada por el usuario, además de que la operación del espectrómetro es controlada desde el ordenador.

Palabras Clave- espectrometría, ultravioleta, instrumentación.

I. INTRODUCCIÓN

El diseño de un espectrómetro hoy en día ha cambiado desde el invento del mismo por Joseph von Fraunhofer en 1814. El primer espectrómetro consistía en un prisma que separaba la luz blanca en sus diversas longitudes de onda, un teodolito, y el ojo humano como sensor. En 1994 la empresa norteamericana Ocean Optics lanzó al mercado el primer espectrómetro comercial en miniatura, el modelo S1000-E. Este se insertaba en una ranura de una computadora personal, utilizaba un detector CCD (Charged Couple Device) lineal y su resolución espectral era de 0.7 nm. Una importante ventaja de este fue la desaparición de partes móviles, todo el rango espectral era capturado al mismo tiempo por el detector de arreglo lineal, cuyas celdas constituían el plano focal del espectrómetro [1]. Para ordenar las diferentes componentes que se diseñaron para el prototipo de espectrómetro propuesto en este documento, los englobamos en las siguientes secciones. El software desarrollado para el funcionamiento y control del espectrómetro se discute en la sección II. Lo relativo a la óptica que se construyó para el instrumento se describe en la sección III. Los resultados de algunas mediciones realizadas por el espectrómetro se muestran en la sección IV. Y por último las conclusiones obtenidas se presentan en la sección V.

II. SOFTWARE

A. Programación en Quartus II.

Quartus II es una herramienta de software producida por ALTERA para el análisis y la síntesis de diseños realizados en HDL (Hardware Description Language). Quartus II permite al desarrollador o desarrolladora compilar sus diseños, realizar análisis temporales, examinar diagramas RTL (Registrer-Transfer Level) y configurar el dispositivo de destino con el programador [2]. La programación del FPGA fue desarrollada en Quartus II a través de VHDL, el

funcionamiento general del programa se describe en el diagrama de flujo figura 1.

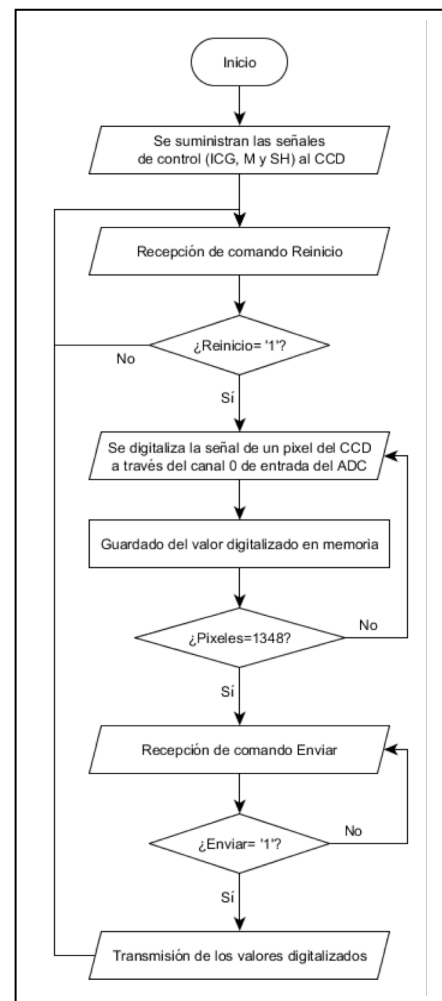


Figura 1. Diagrama de flujo del programa realizado en Quartus II.

B. Programación en LabVIEW

LabVIEW es un software de ingeniería de sistemas que requieren pruebas, medidas y control con acceso rápido a hardware y análisis de datos. El entorno de programación de LabVIEW simplifica la integración de hardware para aplicaciones de ingeniería. LabVIEW reduce la complejidad de la programación [3]. Se desarrolló el Instrumento Virtual Espectro en LabVIEW como programa principal para controlar el espectrómetro, en orden de lograrlo, las tareas que en general realiza son mostrar el espectro, analizarlo y

guardar la lectura de espectro deseada, esto se describe en la figura 2.

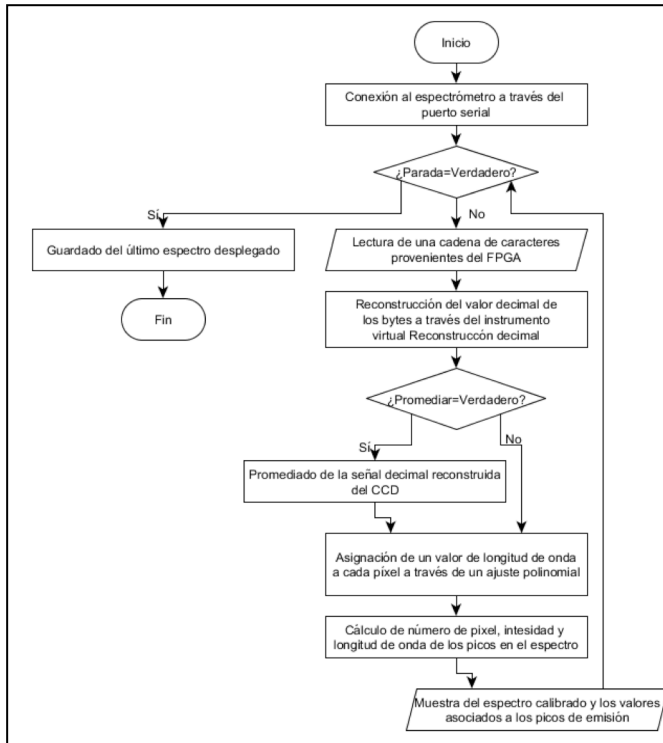


Figura 2. Diagrama de flujo del Instrumento Virtual Espectro.

III. ÓPTICA

A. Banco óptico

Para experimentar con las diversas posibilidades de armar el banco óptico utilizamos una mesa óptica sobre la que montamos los componentes, buscamos que el arreglo óptico permitiera disponer el detector CCD de tal forma que solamente captara la luz proveniente del espejo de enfoque, puesto que la luz extraviada era un problema presente en varios arreglos ópticos que se diseñaron, además de que el espectro visible de una fuente de luz blanca, se viera sobre el plano del detector nítido y brillante, significando esto que el espectro estaba bien enfocado y que se captaba una buena cantidad de luz. Para poder ajustar el espectrómetro a diversas secciones del espectro, es necesario rotar la rejilla de difracción pequeños ángulos, sin necesidad de mover el resto de los componentes ópticos, siempre que se mueve un componente óptico es necesario realizar una nueva calibración cuyo procedimiento se discutirá más adelante.

El banco óptico se instaló dentro de una caja impresa en 3D para proteger y fijar los componentes ópticos, además de por supuesto protegerlo de la luz ambiental indeseada, que pueda generar errores en las mediciones que realiza el instrumento. La caja se diseñó de tal forma que mantuviera unas dimensiones pequeñas, de tamaño similar a los espectrómetros miniatura que se encuentran disponible comercialmente, de tal forma que las dimensiones finales fueron 144.8mm×116.0mm×79.0mm. La electrónica quedó por la parte de afuera instalada en la tapa superior de la caja, con excepción del CCD. Las piezas fueron adheridas a la

caja por medio de tornillos y pegamento, por lo que necesitará un cuidadoso mantenimiento. Todo lo anterior se puede lograr observar en la figura 3 que se muestra a continuación.

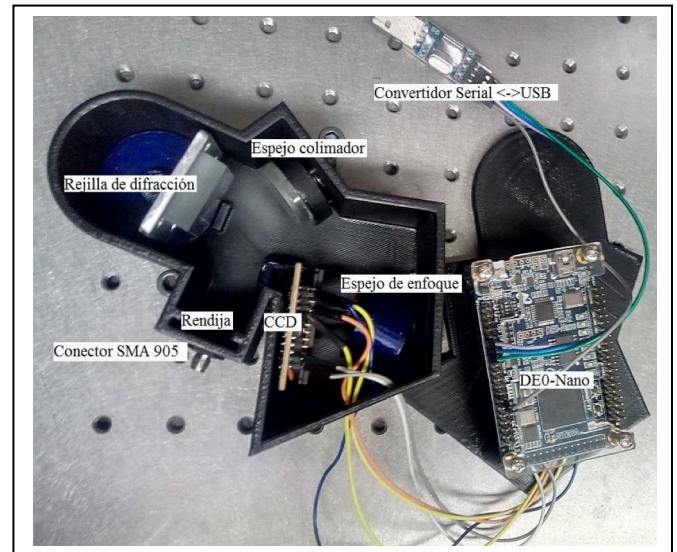


Figura 3. Caja impresa, conteniendo el banco óptico.

B. Calibración

Los pasos para calibrar el espectrómetro se describen a continuación:

1. Capturar la luz proveniente de la lámpara de mercurio a través del cable de fibra.
2. Abrir el instrumento virtual Espectro y ejecutarlo, para visualizar los diversos picos que conforman el espectro de la lámpara en el indicador gráfico Espectro.
3. Usar el botón Promediar para estabilizar la lectura del espectro.
4. Utilizar los controles Umbral de intensidad y Ancho de pico para encontrar los números de pixel de todos los picos que se observan en el espectro.
5. Detenemos la ejecución del instrumento virtual Espectro.
6. Cambiamos los valores de las constantes de calibración en el Diagrama de bloques del instrumento virtual Espectro, asegurándonos de que cada longitud de onda conocida corresponda certeramente a su número de pixel.

El instrumento virtual Espectro fue programado de tal forma que la calibración solamente está considerada para una lámpara de mercurio y 4 de sus picos, los correspondientes a las longitudes de onda: 366.3, 404.4, 435.8 y 546.1 nanómetros, esto se diseñó así para facilitar la calibración del espectrómetro.

IV. RESULTADOS

A. UVP Pen-ray 90001201

Esta lámpara se utilizó para calibrar el espectrómetro. Para conocer el desempeño del espectrómetro construido, lo comparamos con un espectrómetro comercial. En la figura 4 podemos apreciar las mediciones realizadas por ambos espectrómetros, es fácil notar que los picos de emisión

detectados por el espectrómetro que diseñamos, son más anchos que los detectados por el comercial, lo que apunta a una menor resolución espectral por parte del instrumento propuesto. En la tabla 1 se encuentra la comparación de las longitudes de onda medidas que componen el espectro.

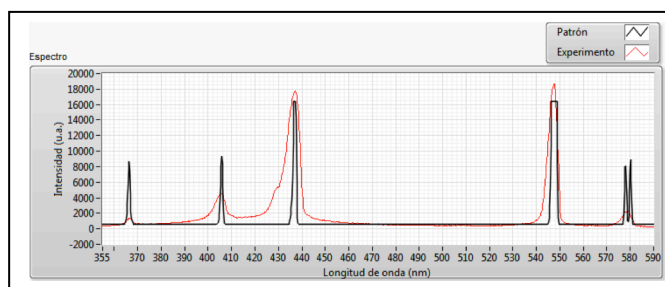


Figura 4. Comparación del espectro de la lámpara de mercurio medido en espectrómetro construido (experimento), y el espectrómetro comercial (patrón), ajustado en escala.

Tabla I
MEDICIONES DE LÁMPARA DE MERCURIO

Fabricante (nm)	Comercial (nm)	Experimental (nm)
253.7	254.7	No detectado
313.2	314.1	No detectado
366.3	366.3	366.6
404.4	405.8	405.4
435.8	437	437
546.1	547.9	547.3
577	578.1	578.6
579.1	580.1	No detectado

B. Láseres

También usamos láseres para continuar comparando espectrómetro armado con uno comercial, puesto que su espectro es cuasi monocromático, además las empresas que los fabrican brindan información sobre su longitud de onda.

Se analizó un láser morado (figura 5) y un láser verde (figura 6), en ambos espectros se puede observar que las mediciones realizadas por nuestro instrumento, tiene un ensanchamiento en los picos de emisión, lo que apunta nuevamente a una resolución espectral menor por parte del instrumento diseñado.

La tabla II recoge las longitudes de onda determinada por ambos espectrómetros

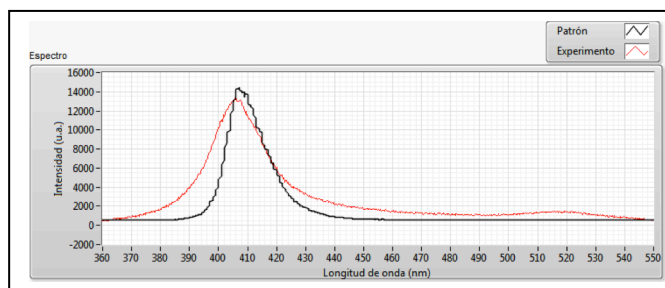


Figura 5. Comparación de láser morado medido en espectrómetro construido (experimento), y el espectrómetro comercial (patrón), ajustado en escala.

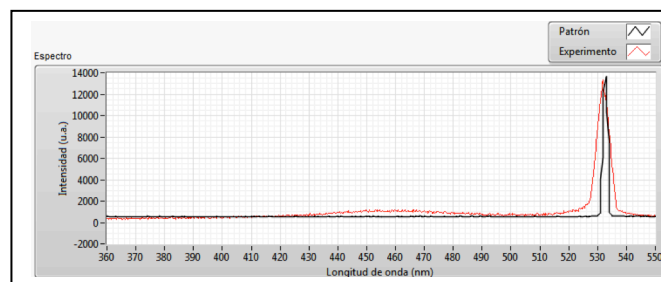


Figura 6. Comparación de láser verde medido en espectrómetro construido (experimento), y el espectrómetro comercial (patrón), ajustado en escala.

Tabla II
MEDICIONES DE LÁSERES

Fabricante (nm)	Comercial (nm)	Experimental (nm)
405	407	404.5
533	533	533.8

C. Resolución espectral

Para determinar la resolución espectral, calculamos la cantidad de nanómetros que le corresponden a cada pixel, simplemente dividiendo el rango espectral entre los 1348 pixeles activos, y además el número de pixeles necesarios para identificar un pico de emisión en la anchura a media altura, encontramos una resolución de 3.3 nm. Al utilizar la región de 530 a 600 nm del espectro de la lámpara de mercurio observada en la figura 7, es fácil ver que los picos del 577 y 579.1 nm no son resueltos por el espectrómetro que construimos, se aprecia como si fuera solamente un pico de emisión centrado en el 578.6 nm, por lo tanto es congruente con nuestra resolución de 3.3 nm, porque para resolver los picos es necesario una resolución de por lo menos 2.1 nm.

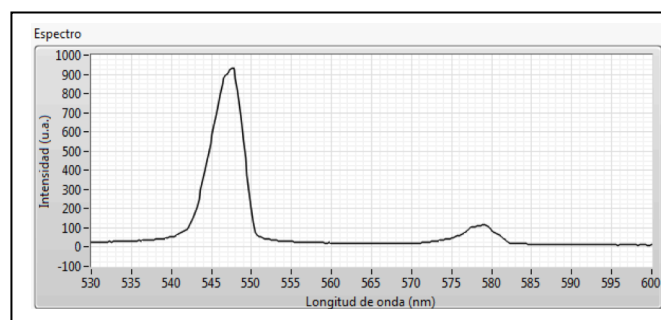


Figura 7. Región de 530 a 600 nm del espectro de la lámpara de mercurio, medido en el espectrómetro construido.

V. CONCLUSIONES

Se diseñó el software en VHDL para programar un FPGA que controla una cámara CCD, digitaliza la señal proveniente de sus primeros 1348 pixeles por medio de un convertidor ADC, transmitir y recibir información a través de un Convertidor Serial ↔ USB hacia una computadora personal.

Se desarrolló un banco óptico impreso en 3D para las posiciones de los elementos ópticos de un espectrómetro (rendija, espejos, rejilla de difracción), el banco tiene las características de ser compacto y ajustable en el rango espectral. De tal forma que facilita la construcción del espectrómetro.

Se construyó un espectrómetro óptico comunicado por USB con un rango espectral de 333.1 a 660.5 nm y una resolución espectral de 3.3 nm, utilizando una cámara CCD controlada por un FPGA.

REFERENCIAS

- [1] Perez Moret, Yohan. (2012) Diseño y construcción de un espectrómetro óptico por USB. LULU.
- [2] Quartus II. Recuperada Noviembre 3, 2017 desde https://es.wikipedia.org/wiki/Quartus_II
- [3] LabVIEW. Recuperada Noviembre 3, 2017 desde <http://www.ni.com/es-mx/shop/labview/buy-labview.html>

Implementación de sistemas bioinspirados

L. F. De La Rosa García^a, L. J. Ontañón-García^b

Coordinación Académica Región Altiplano Oeste

Universidad Autónoma de San Luis Potosí

luisfdlr97@outlook.com^a, luis.ontanon@uaslp.mx^b

Resumen- Gracias a las características de los sistemas dinámicos, se busca representar modelos matemáticos como lo son la neurona de FitzHugh y el modelo de célula β de Pernarowski, en implementaciones físicas utilizando amplificadores operacionales. El objetivo es utilizar este tipo de prototipos, para estudiar y analizar el comportamiento de los modelos en busca de posibles alternativas para realizar estudios de sistemas biológicos sin ser un peligro para los seres vivos.

I. INTRODUCCIÓN

A lo largo de los años el deseo de poder predecir el comportamiento de ciertos fenómenos, ha fascinado a los científicos, grandes científicos como Rössler, Lorenz [1], Lyapunov entre muchos otros dedicaron sus vidas al estudio de Los sistemas dinámicos lo que dio como resultado multiples descubrimientos como lo fue el sistema del clima de Lorenz o el exponente de Lyapunov que nos ayuda a determinar la presencia de caos en un sistema dinámicos. Después de estas aportaciones muchos científicos continuaron desarrollando propuestas de aplicaciones de la teoría del caos para la solución de ciertos problemas e incluso la predicción de eventos en sistemas con múltiples parámetros cambiantes, así como también la adaptación de ciertos modelos a eventos en la vida real, una de estas aplicaciones la de sistemas bioinspirados, la cual trata de replicar el funcionamiento de elementos biológicos basados en ecuaciones diferenciales que modelan su comportamiento.

Tal como hicieron Hodgking y Huxley al estudiar las propiedades eléctricas del axón de un Loligo (Calamar gigante) [2], lograron determinar las propiedades que este tenía de acuerdo al potencial eléctrico del mismo. En la presente investigación se realizó la implementación física del modelo matemático de una neurona realizado por FitzHugh y del modelo matemático de la célula β [3] que reacciona a los impulsos eléctricos del cuerpo, esto podría llevar a alternativas para tratar la diabetes, además de ampliar los conocimientos sobre este tipo de sistemas análogos.

II. DESARROLLO

Para llevar a cabo la implementación física de los modelos, primeramente realizamos la simulación del sistema de ecuaciones y mediante el método Runge-Kutta se evalúa para obtener la solución a manera de gráfica, de esta manera

tenemos una referencia del comportamiento que tendrá el sistema. Después debemos darle solución mediante computo analógico para lo cual se emplearon amplificadores operacionales, este puede colocarse en diferentes configuraciones dependiendo de la operación que deseemos replicar por lo regular se emplean el diferenciador (Figura 1.) y el integrador (Figura 2.)

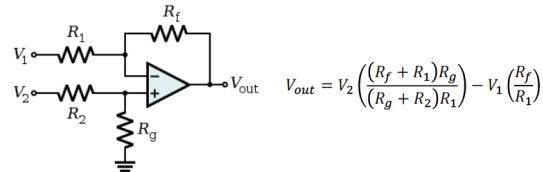


Figura 1. Diagrama y ecuación característica del diferenciador

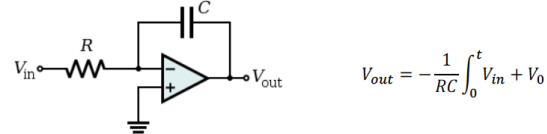


Figura 2. Diagrama y ecuación característica del integrador

En el caso del diferenciador tenemos una resistencia de retroalimentación R_f la cual nos ayuda a determinar el valor que tendrá cada uno de los valores de entrada, este determina por R_f/R_{input} por lo que si deseamos obtener el valor de la resistencia de entrada realizamos lo siguiente $R_f/Valor$, debemos considerar que debido a la ganancia inversora que se obtendrá por el integrador es necesario invertir los signos de la ecuación para obtener un valor correcto. Para elegir la resistencia R_f tomamos un valor de resistencia que sea fácilmente dibicible entre los valores de entrada, debemos tener en cuenta que debe ser de mínimo 10k Ω

$$A_+ = |(a_1) + (a_2)| \quad (1)$$

$$A_- = |(-a_1) + (-a_2)| \quad (2)$$

El cálculo de la resistencia R_g tenemos que saber previamente el valor de R_f y el numero de valores positivos (Ecuación 1) y el número de valores negativos (Ecuación 2) de la ecuación que estamos trabajando, una vez conociendo estos datos podemos emplear Ecuación 3 para determinar el valor de R_g .

$$R_g = \frac{R_f}{A_- + 1 - A_+} \quad (3)$$

Para los parámetros del integrador la resistencia es de $1k\Omega$ y el condensador de $1\mu f$ esto para que la frecuencia del integrador sea $1kHz$. Para los casos en que se debe multiplicar dos valores constantes empleamos un multiplicador de estado AD633AN de igual forma para elevar los valores a una potencia se realiza mediante un arreglo de este tipo de circuitos. Se debe realizar un diferenciador para cada una de las ecuaciones del sistema y la salida de este se conecta al integrador, en este tipo de modelos es muy común encontrarse con que la salida de una ecuación es parámetro de la siguiente y varias casos similares por lo que hay que tener precaución al momento de realizar la conexión. Una vez que se calcularon todas las resistencias se realiza la conexión con ayuda del software de Multisim esto con el fin de obtener una simulación de nuestra implementación, una vez determinado que el sistema si se comporta de acuerdo a la simulacion, se procede a la implementación física teniendo en cuenta que los valores de las resistencias deben ser lo más precisas posibles para obtener un mejor resultado.

III. RESULTADOS

La primera implementación que se realizo fue el modelo neuronal de FitzHugh Nagumo en la cual se sustituyeron los valores constantes marcados anteriormente y se simplifico algebraicamente para determinar la Ecuación 4.

$$\begin{aligned}\dot{x} &= x - \frac{1}{3}x^3 - y + 0.5, \\ \dot{y} &= \frac{1}{12.5}(x + 0.7 - 0.8y),\end{aligned}\quad (4)$$

Para lo cual se emplearon dos multiplicadores de estado (AD633AN) y dos amplificadores operacionales (TL084ACN) (Figura 3) en este caso se implementó con ayuda de la plataforma Elvis II de National Instruments la cual nos permite analizar las señales las cuales se muestran en la Figura 4. Podemos comparar con la simulación del modelo matemático y darnos cuenta que las señales son muy similares por lo que podemos determinar que el modelo se comporta igual que la simulación matemática y que es totalmente posible replicarla en la vida real con pocos componentes electrónicos.

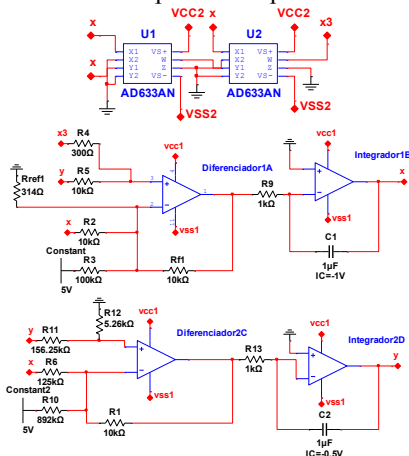


Figura 3. Implementación de la Neurona de FitzHugh

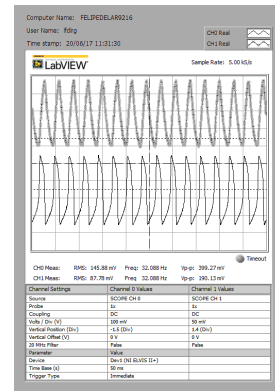


Figura 4. Señales obtenidas de la implementación de la neurona de FitzHugh mediante la plataforma Elvis II NI.

Para la implementación de la célula β se consideró la Ecuación 4 que representa la célula en estado de activación. Se emplearon 2 amplificadores operacionales (TL084ACN) y dos multiplicadores de estado (AD633AN) se consideraron tres diferenciadores uno por cada ecuación del sistema (Figura 5).

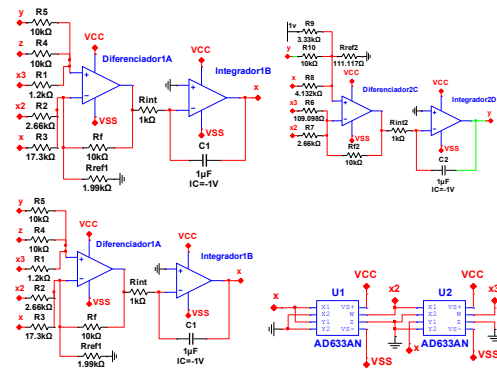


Figura 5. Circuito para la implementación de la célula β

Una vez realizada la simulación del circuito se simuló mediante el software de Multisim (Figura 6) por lo que al determinar que el circuito realmente era funcional se procedió a la implementación física, sin embargo, tras realizar el alambreado el circuito no presentaba la oscilación que debería se comportaba diferente a lo esperado por lo que se utilizaron resistencias lo más precisas posibles sin obtener mucha mejoría, por lo que para analizar el problema se decidió realizar el diagrama bifurcación del sistema para comprender mejor que era lo que estaba ocurriendo con el sistema el cual aun se esta trabajando.

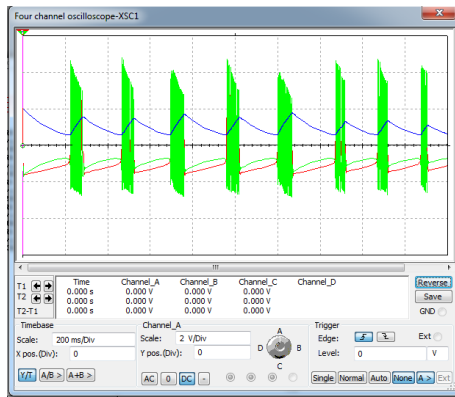


Figura 6. Simulación del circuito en Multisim

IV. CONCLUSIONES

Los sistemas dinámicos nos permiten replicar con mucha precisos fenómenos complejos de la vida real, sin embargo, debemos tener muy en cuenta la aplicación que les vayamos a dar, ya que bajo ciertas circunstancias los sistemas no generan el caos que los caracteriza, por lo que es muy importante tomar en cuenta los valores iniciales y las constantes a utilizar, cabe mencionar que las implementaciones con circuitos físicos no son fáciles de implementar debido a ciertos valores imprecisos que manejan todos los dispositivos de electrónica y que siempre existirán variaciones entre las simulaciones y los resultados que existen realmente.

En el caso de la célula β es necesario encontrar un modelo que se adapte mejor a las características ambientales ya que el que se desarrollo es muy susceptible a los factores ambientales por lo que cualquier señal de ruido genera que no se comporte como debería por lo que una alternativa seria agrandar el circuito para evitar divisiones de voltaje y perdidas de voltaje. Las bifurcaciones son una de las mejores formas de comprender realmente cómo se comporta el sistema y en que intervalos realmente existe caos. Estos nos ayudaran a poder determinar mejor los parámetros a utilizar en el sistema.

La implementación correcta de este sistema tiene un gran potencial debido a que se podría estudiar el comportamiento que tiene este tipo de células en las diferentes etapas de la diabetes e incluso poder llegar a una implementación que ayude a este tipo de células a funcionar correctamente.

REFERENCIAS

- [1] Edwards C. H., Penney D. E. Ecuaciones diferenciales y problemas con valores en la frontera. Pág. 439 y 440. 4ta Edición. México 2009. ISBN: 978-970-26-1285-8
- [2] Hodking, A. L., & Huxley, A. F. (1952). The dual effect of membrane potential on sodium conductance in the giant axon of Loligo. The Journal of physiology, 116(4), 497-506.
- [3] Ontañón-García L.J, Campos-Cantón E., Sánchez-Sahaugun J., Campos-Cantón I., (Octubre 2014). Electronic implement of a pancreatic β cell. XVI Congreso Latinoamericano de Control Automático AMCA, Cancún, Quintana Roo.

Filtro DB

L. F. De La Rosa García^a, D. M Quiroz Moreira^b, B.M Rodriguez Ovalle^c, A. García Paisano^d

Coordinación Académica Región Altiplano Oeste

Universidad Autónoma de San Luis Potosí

luisfdlr97@outlook.com^a, dulcemariaqm97@gmail.com^b, melis-bm-23@hotmail.com^c, garoma056@hotmail.com^d

Resumen- En búsqueda de un método que permita optimizar el procesamiento de la información en imagen, se desarrollo un algoritmo que permite comprimir una imagen tomando los datos de cada pixel y juntándolos en un valor que nos permita minimizar la cantidad de datos a procesar, lo cual puede ser aplicado de forma inversa para recuperar la imagen, así mismo, se debe tomar en cuenta que existen perdidas de datos debido al algoritmo que se utiliza.

NIVEL	INTERVALO	Val. Escalacion
1	0-36	18
2	37-73	55
3	74-110	92
4	111-147	129
5	148-184	166
6	185-221	203
7	222-255	240

Tabla 1. Distribucion de los niveles de escalacion

I. INTRODUCCIÓN

La manipulación de archivos multimedia como imágenes y video es una actividad que en la actualidad se emplean en múltiples ámbitos como imágenes medicas, imágenes militares o bien simplemente el entretenimiento, con tanta información moviéndose a través de internet, se necesita de métodos que permitan optimizar la transferencia de la información ya que los usuarios de dispositivos tecnológicos buscan realizar sus tareas de la forma más rápida y eficiente posible, además hay que agregarle que la seguridad de esta información es primordial para evitar la filtración de información y pérdida de datos. El procesamiento de imágenes es un conjunto de técnicas que permiten cambiar la información de una imagen manipulando la información de los pixeles que la componen[1]. Es aquí donde los métodos de cifrado y de compresión de imágenes toman su importancia ya que mediante la implementación de algoritmos podemos generar un flujo de datos más rápido y garantizar que la información se mantendrá segura para los usuarios destinados.

II. DESARROLLO

El método que se plantea utilizar surgió de un intento de desarrollar un algoritmo para convertir una imagen RGB a escala de grises y poder revertirlo para obtener una imagen a color, sin embargo, tras varios análisis y pruebas se determinó que aunque el algoritmo sí puede llegar a convertir una imagen a escala de grises no puede invertirse el proceso debido a la pérdida de datos que este genera.

El método consiste básicamente en tomar un píxel y determinar un nivel al cual se puede escalar estos niveles fueron determinados dividiendo los 255 en 7 (Tabla 1) ya que estos pueden ser representados por tres bits después de hacer la escalación se agrupan los valores de cuatro píxeles en uno.

Apoyándonos de la herramienta de software Labview 2016 se aplicó el algoritmo, gracias a las herramientas que incluye este software podemos tomar cada píxel de la imagen y descomponerlo en valores binarios es aquí donde realizamos la escalación lo que nos generará un valor de nivel el cual puede ser representado con tres bits, de este modo podemos usar esos tres bits para generar un nuevo valor de píxel en conjunto con los otros 3 píxeles. De este modo el valor de cuatro píxeles queda almacenado en uno solo con un valor totalmente diferente es aquí donde se pueden aplicar métodos de cifrado de manera más óptima.

III. RESULTADOS

Debemos considerar que al pasar una imagen por este algoritmo es inevitable la pérdida de datos, para ciertos valores el error que se produce puede llegar a ser del 50%, aunque podemos considerar que para el ojo humano es muy imperceptible este cambio, es necesario desarrollar algún método que permita al algoritmo comportarse de mejor manera a esta situación.

Al generar la nueva imagen con los valores de píxeles generados se genera una imagen distorsionada, con lo que se pensaría que tiene cierto nivel de cifrado, sin embargo, esto no es así ya que fácilmente se puede extraer la imagen por lo que si se quisiera darle más seguridad a la imagen sería conveniente emplear algún método de cifrado que sí permita la ocultación de forma segura de la imagen.

IV. CONCLUSIONES

El filtro cuenta con la ventaja de que reduce el tamaño de la imagen en una cuarta parte por lo que la cantidad de datos a procesar es mucho menor, reduciendo así el tiempo de procesamiento, sin embargo, al pasar por el filtro el nuevo

pixel de la imagen que va formando el algoritmo es de un valor totalmente diferente a los que lo componen por lo que no es aplicable como metodo de comprecion de imágenes, simplemente puede ser utilizado como un modulo auxiliar para reducir el tiempo de procesamiento. Esto es sumamente importante al momento de transmitir video en tiempo real ya la cual es una de sus principales aplicaciones. Debemos considerar la desventaja que para ciertos valores de pixeles genera un error demasiado grande, esto es mas vsible en imágenes monocromaticas pero aun son necesarias mas pruebas para averiguar en que casos es conveniente utilizarlo y ademas de mejorar la reaccion que tiene el algoritmo a este tipo de condiciones.

REFERENCIAS

[1] González, R.C., Wintz, P.(1996), Procesamiento digital de imágenes. Addison- Wesley, Tema 3,4, páginas 89-269.

Diseño y construcción de una máquina de control numérico

D. Gallegos Baez¹ F. Sánchez Niño¹

¹Universidad Tangamanga Campus Industrias, San Luis Potosí S.L.P, México

Resumen. Se diseñó y construyó una máquina de control numérico (CNC) que permite controlar la posición de un elemento físico, normalmente llamado herramienta que está montada en el efector final del CNC. Mediante el uso de un software que se comunica con el hardware de control del CNC se controlan las coordenadas para posicionar la herramienta de trabajo, estos desplazamientos se realizan siempre respecto a una coordenada de origen (0,0,0) llamada home, es decir, una máquina de control numérico es un sistema de posicionamiento de gran precisión.

El CNC controla todos los movimientos de la herramienta cuando se está manufacturando una pieza, además de realizar el control de posición de la herramienta de trabajo, también controla la velocidad, aceleración y algunos otros parámetros de desplazamiento. Un CNC es un equipo automatizado totalmente integrado, que tiene la capacidad de realizar procesos de corte, grabado, soldadura, barrenado, etcétera.

Índice de términos.- CNC, Arduino, Lenguaje de programación G

I. INTRODUCCIÓN

La máquina de control numérico ha sido uno de los más importantes desarrollos en manufactura en los últimos 50 años [1], [2], al desarrollar:

- Nuevas técnicas de producción
- El incremento la calidad de los productos
- La reducción de costos

Algunas de las ventajas de utilizar un equipo CNC es el fabricar prototipos precisos, cumplimiento de especificaciones, reducción de la dificultad para manufacturar partes, mejorarla planeación de las operaciones, incremento de la flexibilidad de maquinado, reducción en tiempo de programación, mejoramiento de control del proceso y tiempos de maquinado, disminución en los costos por herramientas, incremento de la seguridad para el usuario y reducción del manejo de la pieza de trabajo, aumento de productividad y aumento en la precisión [1], [2].

Las aplicaciones de una máquina de control numérico son:

- Fresado
- Torneado
- Taladrado
- Esmerilado
- Doblado
- Máquina por descarga eléctrica (EDM)
- Inspección

En la tabla 1 se muestra una breve historia del desarrollo del CNC, [1], [2]:

1725	Máquinas de tejer construidas en Inglaterra, controladas por tarjetas perforadas.
1863	M.Forneaux. Primer piano que tocó automáticamente.
1870 – 1890	Eli Whitney. Desarrolló una plantilla y dispositivos de manufactura de partes intercambiables.
1880	Introducción de una variedad de herramientas para el maquinado de metales.
1940	Inicia la investigación y desarrollo del control numérico. Se inician experimentos de producción a gran escala con control numérico.
1955	Las herramientas automatizadas comenzaron a aparecer en las plantas de producción para la Fuerza Aérea de los Estados Unidos.
1956	Hay concentración en la investigación y desarrollo del control numérico.

Tabla 1. Se muestra una breve descripción del avance histórico de la máquina de control numérico.

II. DESARROLLO

En la figura 1 se muestra el diseño de la máquina de control numérico, que consiste de dos ejes de posicionamiento y un eje de trabajo que soporta la herramienta.

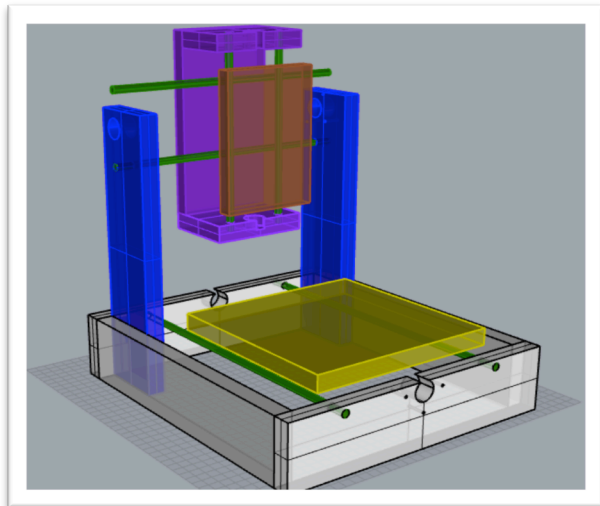


Figura 1. Se presenta el diagrama 3D del diseño de la máquina de control numérico.

A continuación se presentan las especificaciones de diseño del sistema CNC:

- 3 motores paso a paso bipolar Nema17 / 12V / 0,4A / 1,8° por paso.
- 3 acoples flexibles 8mm - 5mm.
- 3 tornillos D8mm- L30cm/ 8mm*giro + tuerca+camisa+m3.
- 12 rodamientos lineales de 8mm.
- 5 ejes acerados 8mm de 40cm.
- 2 soportes de eje de 8mm.
- 1 shield CNC + (3) A4988.
- 1 Fuente 12V 2A.
- 1 soporte mototooldremel 3000.
- 4 sujetadores cama fresado.
- 1 Arduino uno R3.
- 3 rodamientos 8mm.
- 15 cables jumper hembra-macho.
- 20 tornillos para madera 40mm.
- 24 tornillos M4 x 25mm.
- 12 tornillos M3 x 30mm.
- 1 Dremmel 3000 Mototool.
- 6 tablas MDF 30cm x 7 cm / 18mm.
- 1 tabla MDF 20cm x 20cm / 18mm.
- 1 tabla MDF 20cm x 10cm / 15mm.

- 1 tabla MDF 10cm x 5,5cm / 18mm.
- 1 tabla MDF 13cm x 10cm / 15mm.
- 1 tabla MDF 10cm x 7 cm / 18mm.

En las figuras 2 y 3 se muestra el proceso de ensamble de la estructura de la máquina de control numérico.

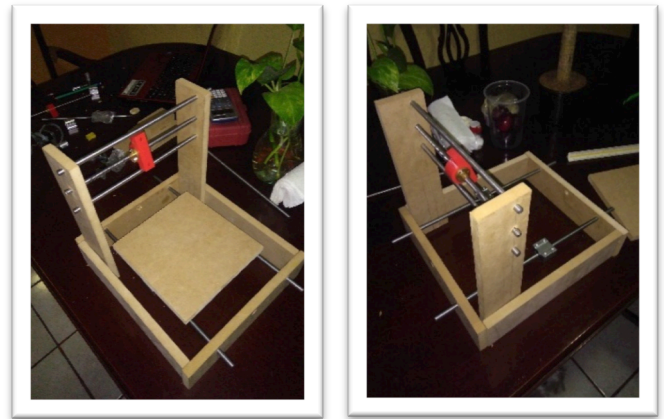


Figura 2. Se muestra el ensamble de los ejes "x", "y" y "z" del sistema CNC.

Se utilizó madera para construir cada uno de los ejes de la máquina de control numérico, que se desplazan a lo largo de guías acopladas a motores paso a paso. Los motores se energizan a través del driver L293D que es controlado por el Arduino, que recibe las coordenadas de desplazamiento desde el programa de la computadora.



Figura 3. Los ejes "x", "y" y "z" se desplazan sobre guías acopladas a motores paso a paso que son energizados para realizar el desplazamiento indicado desde el programa de la computadora.

Una máquina CNC es programada en el lenguaje de programación conocido como “G”, el cual indica la acción a realizar del sistema a través de una serie de coordenadas xy (posicionamiento) así como la herramienta a utilizar y la velocidad de trabajo (corte, grabado, barrenado, decapado). En la figura 4 se muestra el diagrama de flujo del proceso de funcionamiento de la máquina de control numérico diseñada.

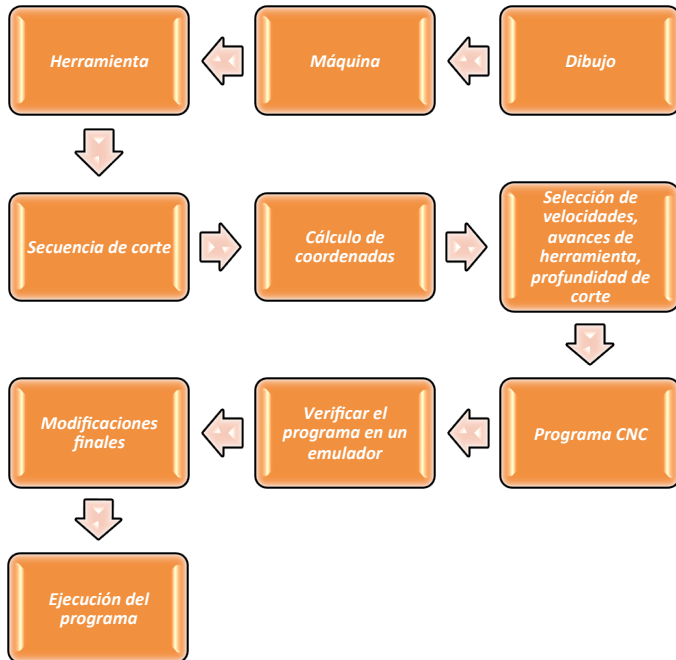


Figura 4. Diagrama de flujo de funcionamiento del CNC.

El funcionamiento del sistema se define de la siguiente forma: se tiene una figura, fotografía o el diseño de un PCB de un circuito electrónico, se vectorizan estos diseños utilizando el programa Inkscape y se obtiene el código G a partir de la figura vectorizada. En el código G se indica el tipo de herramienta a utilizar y se genera la secuencia de corte y coordenadas de posicionamiento. Se ajustan los parámetros de trabajo del sistema y se carga el programa en el Arduino, que activa los motores paso a paso en función del trabajo a realizar definido en el código G.

III. RESULTADOS

En la figura 5 se muestra el diseño final del CNC construido, se tiene un sistema con tres grados de libertad que le permiten posicionarse en el espacio bidimensional a través del reconocimiento de una coordenada xy, una vez ahí la herramienta de trabajo descende para realizar la tarea definida desde el programa de control. Se utiliza el programa Grbl Controller que interpreta el programa en código G para convertirlo en el número de pasos que ejecutarán los motores paso a paso del sistema electrónico de control. Los motores paso a paso tienen una resolución de 200 pasos por revolución,

es decir, el desplazamiento mínimo que pueden realizar es de 1.8°, lo que permite tener movimientos muy precisos en el orden de los milímetros, de esta forma se asegura la obtención de trabajos de calidad.

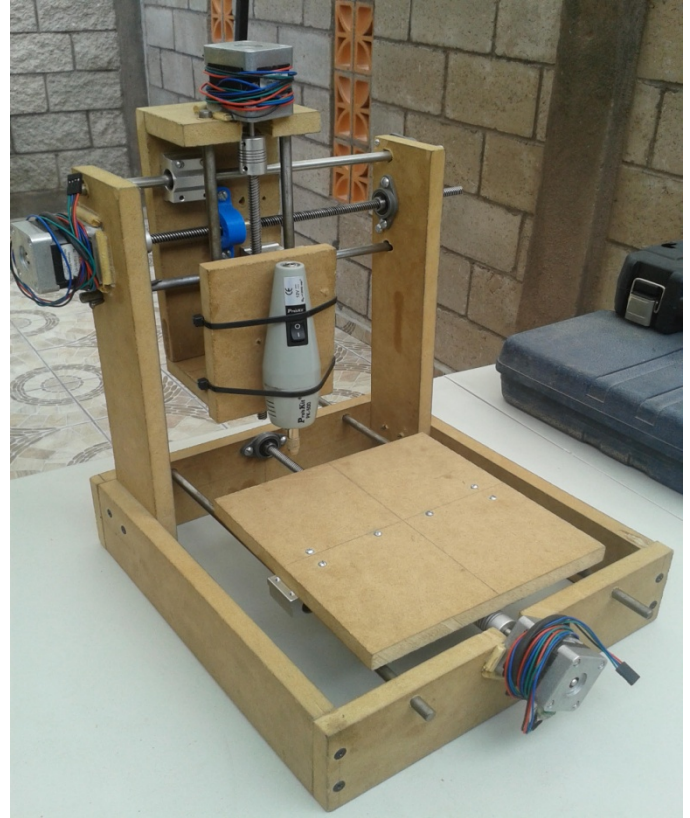


Figura 5. Diseño final de la máquina de control numérico.

En la figura 6 podemos observar el ambiente de trabajo del Grbl Controller.

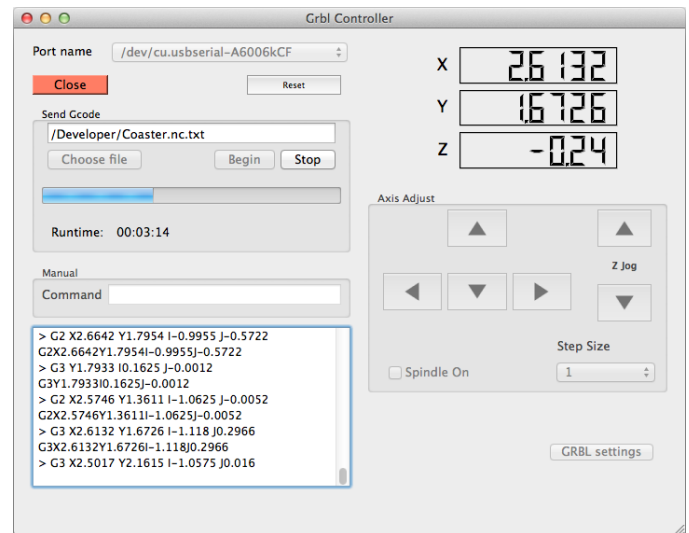


Figura 6. Se muestra la interface de control del Grbl Controller que permite manipular al CNC en modo manual y modo automático.

Desde la interface del Grbl Controller se pueden definir parámetros de trabajo como la velocidad, aceleración, tiempos de espera y de trabajo, se carga el programa en código G obtenido desde el Inkscape y se envían los comandos a través del puerto USB que trabaja en modo serial emulado hacia el Arduino. Por medio del Grbl Controller se puede manipular el CNC en dos formas: automático y manual. El modo manual permite verificar el correcto funcionamiento del sistema así como ajustar el Home y los finales de trayectoria de los ejes. El modo automático ejecuta el programa que permite manufacturar una pieza de acrílico, plástico, madera o una tarjeta de circuito impreso.

IV. CONCLUSIONES

Se dio a conocer una breve introducción sobre la evolución, diseño y construcción de una máquina de control numérico también conocida como CNC. El sistema se construyó en todos sus niveles, mecánica, hardware y software, obteniendo como resultado una máquina funcional para realizar diversas funciones, como puede ser el barrenado, decapado, corte, etcétera, en materiales como madera, plástico, acrílico o tarjetas de circuito impreso. Se tiene contemplado seguir realizando mejoras al sistema para obtener trabajos finales de mejor calidad.

V. REFERENCIAS

- [1]http://www.ubiobio.cl/miweb/webfile/media/106/descargas/CIM/CNC_CIM_2007_David.pdf
- [2]<http://materias.fi.uba.ar/7565/U4-control-numerico-por-computadora.pdf>

Diseño y prototipado de humanoide por impresión 3D.

Israel Macias González
Facultad de Ingeniería
Universidad Autónoma de San Luis Potosí

Juan Almendárez Rodríguez
"Instituto de Investigación en Comunicación Óptica"

Resumen- Las técnicas de impresión 3D están facilitando el desarrollo de robótica de bajo coste, permitiendo el diseño y construcción de robots, aunque no son adecuados en entornos industriales, sin embargo, pueden ser usados en educación, entretenimiento, prototipos, sistemas de robótica sociales, entre otras aplicaciones. En este proyecto se detalla el proceso de construcción, programación y componentes involucrados para el control y puesta en marcha de un robot humanoide con el uso de tarjetas de open-hardware como lo es TI LaunchPad.

INTRODUCCIÓN

Hoy en día, el avance de la robótica se ha convertido en un gran aporte al desarrollo de la sociedad en campos diferentes tales como: la medicina, la industria, el entretenimiento, etc.

La robótica bípeda conforma un área de investigación con gran crecimiento en los últimos años. Si bien el desplazamiento mediante ruedas es más eficiente y permite mayor velocidad, los robots con patas son más versátiles y pueden desplazarse en terrenos irregulares. En particular, los bípedos son esencialmente aptos para manejarse en nuestro entorno, por contar con características similares a la de los seres humanos.

PROBLEMÁTICA A TRATAR

Actualmente para contar con un robot bípedo es necesario comprar kits que en la mayoría de los casos son de elevado costo lo que impide que en muchas escuelas o centros educativos incluyan en su programa académico planes de estudio de ciencia y tecnología aplicados como lo es la robótica.

OBJETIVOS A TRATAR

Este proyecto plantea un estudio sobre los robots bípedos mediante la construcción de un prototipo de bajo coste, pero que sea capaz de realizar tareas relativamente complejas, como desplazarse caminando. Los resultados obtenidos pueden servir de base para trabajos futuros que apunten al desarrollo de robots bípedos y la investigación en esta área.

I. SISTEMA DE CONTROL

La base principal del robot bípedo es la tarjeta de desarrollo esta controla a los 17 servo motores y además permite comunicación bluetooth con alguno otro dispositivo que cuenta con la app de comandos de voz para controlar al robot.

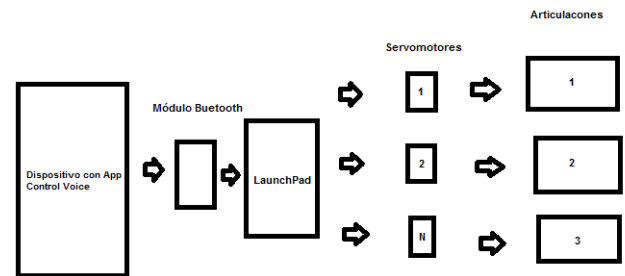


Fig. 1. Diagrama de Bloques del Sistema.

En la Figura 1 se muestra el **Diagrama de Bloques** el cual hace referencia al sistema.

II. COMPONENTES DEL EQUIPO

El proyecto a presentar consta de diferentes secciones por lo cual se presenta de forma detallada en los siguientes párrafos.

A.Estructura.

Todas las partes del robot fueron creadas en el software de dibujo asistido por computadora DS SolidWorks y debido a que el proyecto aún está en la etapa de prototipo la estructura principal está construida de PLA (Ácido Poliláctico) por medio de impresión 3D. En la Figura 2 se aprecia una fotografía del prototipo.

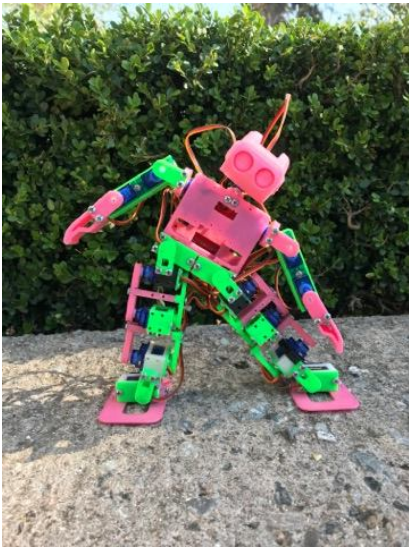


Fig 2. Estructura del prototipo de humanoide.

B. Sección de Hardware.

Las dos secciones principales del robot son la de control que cuenta con tarjeta de desarrollo de Texas Instruments que cuenta con el microcontrolador EK-TM4C123GXL [1] y la sección de baterías y servo motores Sg-90. El robot cuenta con 17 servomotores ya que la tarjeta de desarrollo no entrega la suficiente corriente para alimentar a todos los servomotores se agregaron baterías con el fin de alimentar con la corriente necesaria a todos los servomotores.

En la Figura 3 uno de los servomotores usado para el robot.



Fig 3. Servo Motor Sg-90.

La electrónica desarrollada se basa en el uso de una tarjeta de desarrollo tipo LaunchPad de la empresa Texas Instruments que cuenta con un microcontrolador que cuenta hasta con 36 salidas permitiendo controlar una gran cantidad de servomotores. En la figura 4 se muestra la tarjeta de desarrollo del sistema.



Fig 4. Tarjeta de desarrollo del robot.

C. Sección del Software.

La plataforma de programación del microcontrolador es Energía [2] ya que es un entorno de desarrollo integrado (IDE) y software de código abierto e impulsado por la comunidad. El desarrollo del código comunicación bluetooth y el programa de movimiento del robot se desarrolla en lenguaje C.

En la Figura 5 se parte del código en C de comunicación bluetooth.

```
Comunicaci_n_bluetooth

int estado=1;
void setup(){
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(7,OUTPUT);
  pinMode(6,OUTPUT);
}

void loop()
{
  if(Serial.available()>0)
  {
    estado = Serial.read();
  }

  if (estado == 'a')
  {
    digitalWrite(13,1);
  }

  if (estado == 'b')
  {
    digitalWrite(13,0);
  }
}
```

Fig 5. Código en C que permite la comunicación bluetooth del robot.

Para hacer las rutinas de movimiento del robot fue necesario observar el comportamiento de cada uno de los motores y de ahí establecer límites en cada uno ellos para mantener la estabilidad en el robot.

En la Figura 5 se muestra el código en C de algunas rutinas del movimiento del robot.

```

Programa_de_movimiento_del_humanoide Servo.h
#include <Servo.h>
//PIERNA IZQUIERDA//
Servo servo0;
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
//PIERNA DERECHA//
Servo servo5;
Servo servo6;
Servo servo7;
Servo servo8;
Servo servo9;
//BRAZO IZQUIERDO//
Servo servo10;
Servo servo11;
Servo servo12;
//BRAZO DERECHO//
Servo servo13;
Servo servo14;
Servo servo15;
//CABEZA DEL ROBOT//
Servo servo16;

void setup()
{
  servo0.attach(23);
  servo1.attach(24);
}

```

Fig 6. Código en C de rutinas de movimiento del robot.

Para establecer comunicación inalámbrica con otro dispositivo se usó la app de Control Voice [3] con el fin de controlar las rutinas del robot por medio de comandos de voz. En la figura 7 se muestra el logo de la app Control Voice.



Fig 7. App Control Voice.

III. CONCLUSIONES

Se está diseñando y construyendo de manera exitosa un prototipo de robot humanoide con el uso de SolidWorks, impresión 3D y la tarjeta de desarrollo Launch Pad.

IV. RESULTADOS

En las figuras 8 y 9 se muestra la vista frontal y posterior del robot ya construido.

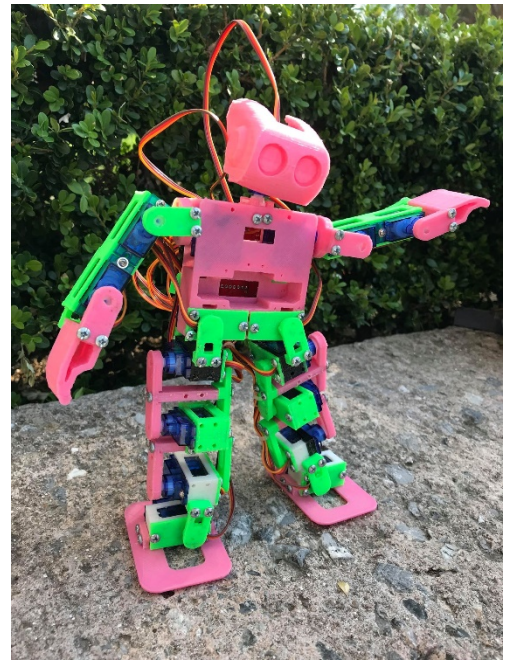


Fig 8. Vista frontal del robot humanoide.



Fig 9. Vista posterior del robot humanoide.

REFERENCIAS

- [1]. Texas Instruments [Internet], (12 de Abril de 2018) Obtenido de <http://www.ti.com/tool/EK-TM4C123GXL>
- [2]. Energia [Internet], (12 de Abril de 2018) Obtenido de <http://www.ti.com/tool/ENERGIA>

Luminaria LED auto recargable sin necesidad de paneles solares

Israel Macias González
Facultad de Ingeniería
Universidad Autónoma de San Luis Potosí

Juan Almendárez Rodríguez
“Instituto de Investigación en Comunicación Óptica”

Resumen- El siguiente proyecto consiste en el desarrollo de una luminaria LED, la cual se puede auto recargar sin la necesidad de paneles solares, ya que desde hace más de 100 años comenzó la idea de aprovechar la luz del sol como una fuente de energía limpia. Hoy en día se cuenta con paneles solares los cuales aún son de elevado costo por lo que el desarrolló esta luminaria permitirá aprovechar la energía solar sin necesidad de uno de ellos.

I. INTRODUCCIÓN

La energía solar-fotovoltaica es una tecnología de generación de energía basada en la conversión de la radiación solar a corriente directa por medio de materiales semiconductores a través del efecto fotoeléctrico; la idea de utilizar y aprovechar este tipo energía data de hace más de 100 años, pero ya que era muy costosa la fabricación de paneles solares y de muy baja eficiencia provocaba que la inversión que se tenía que realizar para implementar un sistema fotovoltaico se redituaba a largo plazo, mayor a 20 o 25 años. Hoy en día la fabricación de paneles solares ha ido en aumento y por lo tanto su precio ha disminuido, llegando a un punto en que la inversión es redituable de 5 a 8 años, de igual forma se ha buscado reducir el consumo de energía de electrodomésticos. Uno de los dispositivos eléctricos de mayor consumo energético en el hogar son los focos o bombillas incandescentes debido a que se encuentran encendidas durante largos periodos de tiempo; actualmente se comienza la implementación de luminarias LED, en las cuales su consumo es menor a comparación de las bombillas incandescentes.

PROBLEMÁTICA A TRATAR

Actualmente para utilizar y aprovechar la energía solar-fotovoltaica es por medio de módulos fotovoltaicos, por lo que se presenta con este proyecto una forma diferente de utilizar y aprovechar la energía por la radiación solar.

OBJETIVOS A TRATAR

El objetivo del presente proyecto es crear una forma nueva para captar la energía basada en la conversión de la radiación solar a corriente directa por medio de materiales semiconductores con el uso de LEDs.

Ya que el alumbrado público es uno de los principales consumidores energéticos durante largos periodos de tiempo, el presente proyecto permitiría captar energía en forma de radiación solar durante el día y aprovecharla durante su periodo de trabajo.

II. SISTEMA

En la Figura 1 se muestra el **Diagrama de Bloques** el cual hace referencia al sistema. La luminaria capta la energía de la radiación solar y la convierte a corriente directa por lo que se encuentra conectada a las baterías por un arreglo de transistores. De igual el voltaje de la luminaria LED es leído por el microcontrolador **EK-TM4C123GXL de Texas Instruments** [1] para la retroalimentación del seguidor solar.

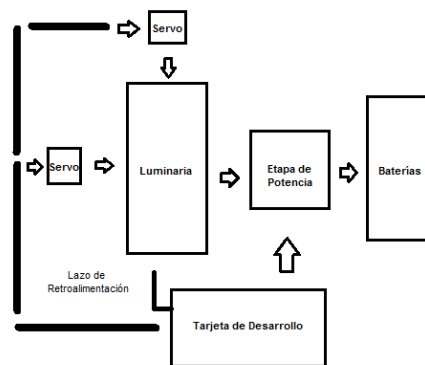


Fig. 1. Diagrama de Bloques del Sistema.

III. SECCIONES DEL SISTEMA

El proyecto a presentar consta de diferentes secciones por lo cual se presenta de forma detallada en los siguientes párrafos.

A.Estructura.

Debido a que el proyecto aún está en la etapa de prototipo la estructura principal está construida de PLA (Ácido Poliláctico)

por medio de impresión 3D, ya que una de las principales áreas de aplicación de la luminaria es alumbrado público se decidió hacer su estructura similar a la que hay en la ciudad de San Luis Potosí.



Fig 2. Estructura mecánica del prototipo.

B. Sección de Hardware.

El prototipo desarrollado emplea baterías para almacenar la energía eléctrica que es recibida por el sol durante el día, mientras que por la noche estas mismas proporcionan la energía para el encendido de los LEDs que simulan la iluminación de una avenida u hogar. La electrónica desarrollada se basa en el uso de una tarjeta de desarrollo tipo LaunchPad de la empresa Texas Instruments que cuenta con un microcontrolador de ultra-bajo consumo de energía, lo que nos permite almacenar una mayor cantidad de energía durante el día, el arreglo de los LEDs y transistores esta diseñada para pasar de estar en una conexión en serie a una conexión en paralelo con las baterías. En la figura 3 se muestra las baterías usadas por el sistema.



Fig 3. Baterías del prototipo.

En la figura 4 se muestra la tarjeta de desarrollo con el microcontrolador **EK-TM4C123GXL**.



Fig 4. Tarjeta de desarrollo del sistema.

De igual forma cuenta con dos **Servo Motor Sg-90** para cada eje de la base y el domo que cuenta con la luminaria LED con el fin de que permitan su movimiento al seguir la trayectoria solar. Tales motores son controlados por el microcontrolador EK-TM4C123GXL. En la figura 5 se muestra un **Servo Motor Sg-90** usado para el sistema.



Fig 5. Servo Motor Sg-90.

C. Sección de Software.

La plataforma de programación del microcontrolador es Energía [2] ya que es un entorno de desarrollo integrado (IDE) y software de código abierto e impulsado por la comunidad. El desarrollo del código del Seguidor Solar como la activación de la etapa de potencia que convierte a la luminaria LED en generadora o consumidora de energía eléctrica fue desarrollado en esta plataforma a través de lenguaje C.

En la figura se muestra parte del código programa del sistema

```

LEDs_Solar_Tracker

#include <Servo.h> //Definiendo Servos

Servo servohorizontal;
int servoh = 0;

int servohlimitMax = 180;
int servohlimitMin = 0;

Servo servovertical;
int servov = 0;
int servovlimitMax = 160;
int servovlimitMin = 0;

//Asignando lecturas de entrada

int Columna1= 4;
int Columna2= 5;
int Columna3= 6;
int Columna4= 7;

void setup ()
{
  servohorizontal.attach(2);
  servohorizontal.write(0);
  servovertical.attach(3);
  servovertical.write(0);

  delay(500);
}

```

Fig 6. Código en C del programa del sistema.

IV. CONCLUSIONES

Se está diseñando y construyendo de manera exitosa un prototipo de seguidor solar con luminaria LED autorecargable dirigido para su uso en el alumbrado público de la ciudad de San Luis Potosí.

V. RESULTADOS

La luminaria a el día de hoy se encuentra en un arreglo de 12 x 12 LEDs, los cuales se encuentran en paralelo en 6 columnas de 12 LEDs como lo muestra la figura.

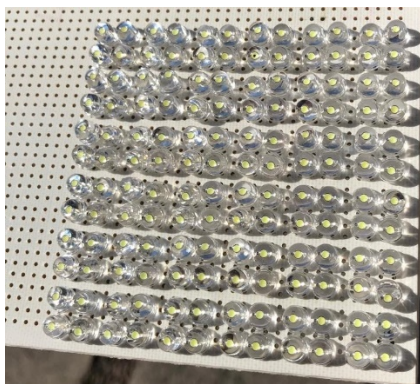


Fig 6. Luminaria LED autorecargable de 12 x 12 LEDs..

En la figura se aprecia el voltaje entregado por una columna, que sería el voltaje entregado por un LED debido a su arreglo con los otros 11 LEDs.



Fig 7. Medición de una sola columna de la luminaria.

En las figuras 8 y 9 se observa la medición de voltaje en diferentes columnas con el uso de multímetro y osciloscopio.

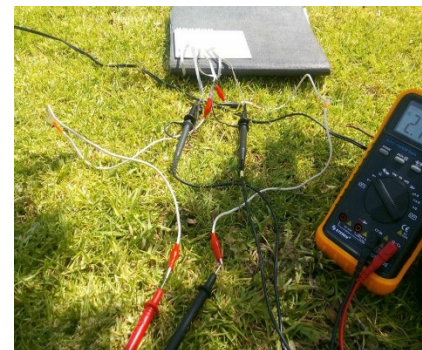


Fig 8. Medición del voltaje de diferentes columnas de la luminaria.

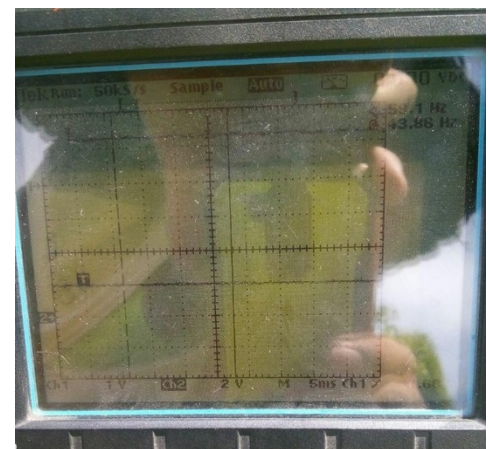


Fig 9. Medición del voltaje de dos columnas con el uso de osciloscopio .

Ha el día de el voltaje máximo que entrega un LED es de 2.224 V, sin embargo el voltaje y la corriente depende de los materiales implementados para cada LED y de su forma de construcción, así como las condiciones climatológicas en las que se encuentra operando, por ello se esta trabajando en un sistema de caracterización de luminarias LED auto recargables con monitoreo de condiciones meteorológicas.

REFERENCIAS

- [1]. Texas Instruments [Internet], (12 de Abril de 2018) Obtenido de <http://www.ti.com/tool/EK-TM4C123GXL>
- [2] Energia [Internet], (12 de Abril de 2018) Obtenido de <http://www.ti.com/tool/ENERGIA>

Robot Autónomo-Teleoperado

Juan Alberto Vértiz Hernández, Lorena Rodríguez Salas, Omar Vital Ochoa

Instituto de Investigación y Comunicaciones Ópticas de la Universidad Autónoma de San Luis Potosí. IICO-UASLP
San Luis Potosí, México

Resumen— Este proyecto presenta una alternativa segura y confiable para el control de un robot autónomo y teleoperado. Tiene la característica de establecer una comunicación vía bluetooth por medio de una aplicación, para sistema operativo Android e iOS, y la programación realizada en lenguaje C para el PIC18F4550 donde se tiene el control de velocidad y dirección de los motores utilizando PWM y un sensor ultrasónico para calcular distancias y evitar que choque y/o caiga al vacío.

Objetivo— Diseñar un robot con orugas controlando sus movimientos con el PIC 18F4550 y una aplicación móvil para su manipulación mediante comunicación bluetooth.

Objetivos Particulares:

1. Diseñar y programar el microcontrolador para manejar los motores y sensor ultrasónico.
2. Diseñar y programar la aplicación en Android e iOS que permita el control del robot en forma autónoma y teleoperada.

Palabras Clave- PIC18F4550, Moto-Reductor, Bluetooth

I. INTRODUCTION

En la actualidad, dentro de la robótica móvil, se ha desarrollado prototipos y modelos robóticos para un sinnúmero de soluciones particulares. Por ejemplo, los robots con orugas tienen diferentes aplicaciones tales como: búsqueda de personas en espacios confinados, detección de minas, generación de mapas topográficos, detección de fallas en canales pluviales, para enseñanza didáctica, entre otros.

Por lo tanto, el proyecto propone diseñar y construir un robot móvil equipado con todos los elementos y herramientas estructurales y de locomoción básicos, que posteriormente, con mejoras en el diseño y equipo puede utilizarse por una persona entrenada para adelantar actividades de búsqueda y exploración en estructuras colapsadas y semicolapsadas o en espacios pequeños donde no acceda alguna persona, teniendo presente lograr reducir todo riesgo y en lo posible al menor costo. De manera que este trabajo presenta el diseño de un prototipo robot con desplazamiento mediante orugas. Operado mediante dos formas: autónomo y teleoperado.

a) *Autónomo:* Con un sensor ultrasónico se detecta la proximidad de obstáculos, haciendo que éste se detenga y cambie de dirección.

b) *Teleoperado:* Utilizando una APP móvil por conexión vía bluetooth se generan los comandos de movimiento.

Gracias a que los avances tecnológicos permiten el uso de diversos hardware y software para poder determinar una tarea específica en el programa utilizado. Tal es el caso de este

proyecto, donde se utilizó el entorno de desarrollo CCS C Compiler, para realizar la programación en lenguaje C y posteriormente se recurre al entorno de desarrollo MPLAB X IDE junto con su interface para programación de tarjeta ICD3, para cargar el código en la tarjeta de desarrollo que contiene el microcontrolador PIC 18F4553.

II. MARCO TEÓRICO

Conforme crece el conocimiento y las diversas necesidades del hombre, éste va creando nuevas herramientas que le permitan la recurrir al uso de un robot, con el fin de que realice una determinada actividad. Se pueden encontrar diversos tipos de robots tales como: robots industriales, móviles, androides, médicos, teleoperadores, poliarticulados, zoomórficos, etcétera. Mismos que requieren de una aplicación directa mediante software y hardware.

A. Software

a) Lenguaje C

Es un lenguaje estructurado, en el mismo sentido que lo son otros lenguajes de programación como el lenguaje Pascal, el Ada o el Modula-2, pero no es estructurado por bloques, o sea, no es posible declarar subrutinas (pequeños trozos de programa) dentro de otras subrutinas, a diferencia de como sucede con otros lenguajes estructurados. Además, el lenguaje C no es rígido en la comprobación de tipos de datos, permitiendo fácilmente la conversión entre diferentes tipos de datos y la asignación entre tipos de datos diferentes [1].

B. Hardware

a) Motor de Corriente Directa

El motorreductor es un potente motor de corriente continua con relación 30:1 y caja de engranajes de metal, destinada a operar a 9V. Estas unidades tienen un eje en forma de D de salida de 0,61" de largo, 6 mm de diámetro [2].



Pololu

C. Puente H

Figura 1. Motorreductor 30:1.

El módulo L298N es un dispositivo que proporciona la corriente necesaria para controlar motores DC o motores a

pasos en aplicaciones robóticas. Si queremos controlar la velocidad de un motor de DC, tenemos que hacer uso de un PWM. Este PWM será aplicado a los pines de activación de cada salida o pines ENA o ENB respectivamente. Por tanto, los jumpers de selección NO serán utilizados. [3].



Figura 2. Puente H, L298N.

D. Microprocesador PIC18F4550

Este microprocesador fabricado por Microchip, posee varias características que lo hacen versátil, eficiente y práctico para ser empleado en aplicaciones de robótica. Algunas de sus características principales son [4]:

- Microcontrolador con módulo USB especificación 2.0. Soporta Low speed 1.5Mb/s y full speed 12Mb/s. 32 endpoints (16 bidireccionales). 1kB de memoria de doble acceso para el USB.
- Hasta 35 pines I/O disponibles.
- Memoria de programa flash de 32 kB.
- RAM de 2048 Bytes.
- EEPROM de datos de 256 Bytes.
- Velocidad de la CPU 12 MIPS.
- Oscilador externo hasta 48 MHz.

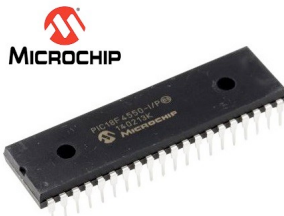


Figura 3. Microcontrolador PIC18F4550.

E. Bluetooth 4.0 y HC-05

Permite la comunicación entre dispositivos de pila de botón y dispositivos Bluetooth, que opera en 2.4 GHz (una de las bandas ISM), con una tasa de transferencia de 1 Mbps en la capa física. Está basado en un microchip de bajo costo con opciones más amplias para su empleo en la industria; además tiene el mismo tamaño de los dispositivos Bluetooth, como los que se encuentra en un teléfono móvil. Tiene soporte para seguridad, ya que emplea el sistema de cifrado AES y esquemas de seguridad configurables [5]. Por su parte el módulo HC-05 es un módulo Bluetooth SPP (Protocolo de

puerto serie) fácil de usar, diseñado para la configuración de conexión en serie inalámbrica y transparente.

El módulo Bluetooth 4.0 del puerto serie está completamente calificado Bluetooth V2.0 + EDR (velocidad de datos mejorada) modulación de 3Mbps con transceptor de radio completo de 2.4GHz y banda de base. Utiliza el sistema Bluetooth CSR Bluecore 04-External single chip con tecnología CMOS y con AFH (función de salto de frecuencia adaptativo). Tiene una huella tan pequeña como 12.7 mm x 27 mm. Espero que simplifique su ciclo general de diseño/desarrollo. Tiene un alcance aproximado de 60 m [6].

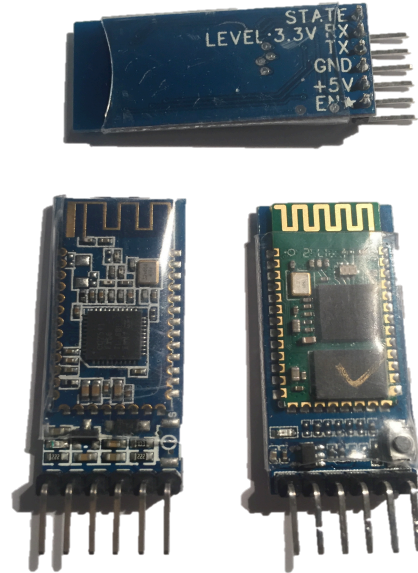


Figura 4. Bluetooth 4.0 (MLT-BT05) y HC-05. En la parte superior se muestran los pines de conexión en formato Serial RS-232, abajo a la izquierda el MLT-BT05 para iOS y a la derecha el HC05 para Android

III. MATERIALES

Para la realización de este proyecto se utilizaron los siguientes materiales:

1. Un puente H L298N.
2. Un sensor ultrasónico SRF06.
3. Un PIC 18F4550.
4. Cuatro baterías de 1.5V y una de 9V.
5. Dos motorreductores de corriente directa de 30:1.
6. Un módulo bluetooth HC-05 para Android y un bluetooth 4.0 para iOS.
7. Dos rodamientos.
8. Dos bandas de hule.
9. Una estructura de plástico.

IV. PROCEDIMIENTO

Para el desarrollo del proyecto, es necesario tener conocimientos de programación en lenguaje C, es decir sus directivas de programación, así como las interfaces del mismo y conocer los componentes que se utilizan [8].

El proyecto se va realizando por pasos debido a que se tienen que ir probando por partes los componentes utilizados. Primero se realiza la programación en el PIC para que pueda enlazarse con el dispositivo bluetooth, ya sea el 4.0 para iOS o el HC-05 para Android. Posteriormente se realiza otro código para el control de velocidad de los motores de CD utilizando PWM. Por último, se realiza la codificación del sensor de proximidad. De manera que el último paso se enfocó en conjugar los códigos previamente hechos en uno sólo programa para poder cargarlo en el PIC mediante la tarjeta de programación.

A. Configuración de los parámetros de control de los motores

Para tener un control óptimo del robot, se programan las direcciones de los motores para lograr cada uno de los movimientos requeridos del robot, así como las velocidades y duración de cada uno de los movimientos. La Tabla 1 muestra los valores digitales requeridos para lograr est, así como los tiempos de operación del PWM de cada uno de los motores.

TABLA I. SECUENCIA PARA EL MANEJO DE LOS MOTORES Y LA VELOCIDAD.

<i>Acción</i>	<i>DIR A (Pin RB0)</i>	<i>DIR B (Pin RB1)</i>	<i>PWM A (CCP1 RC1)</i>	<i>PWM B (CCP2 RC1)</i>
<i>Adelante</i>	1	1	D = 50%	D = 50%
<i>Reversa</i>	0	0	D = 50%	D = 50%
<i>Derecha</i>	1	0	D = 50%	D = 50%
<i>Izquierda</i>	0	1	D = 50%	D = 50%
<i>Detener</i>	x	x	D = 0%	D = 0%

a. Parámetros de Control de los Motores y Velocidad.

a) Esquemático del circuito electrónico.

En la figura 5 se muestra el diagrama esquemático del alambrado para el Proyecto. La figura muestra el cableado que se tiene entre el Puente H con los motores, el puerto B y el puerto C de la tarjeta de programación del microprocesador PIC18F4550, para la manipulación de del giro del motor. Es importante mencionar que los motores se programaron en 4 direcciones:

1. Los dos motores se programan con dirección al frente.
2. Los dos motores se programan con dirección hacia atrás.
3. Para dar vuelta a la izquierda el motor derecho gira al frente y el derecho gira hacia atrás.
4. Para que gire a la derecha se programa el motor izquierdo hacia atrás y el derecho hacia al frente.

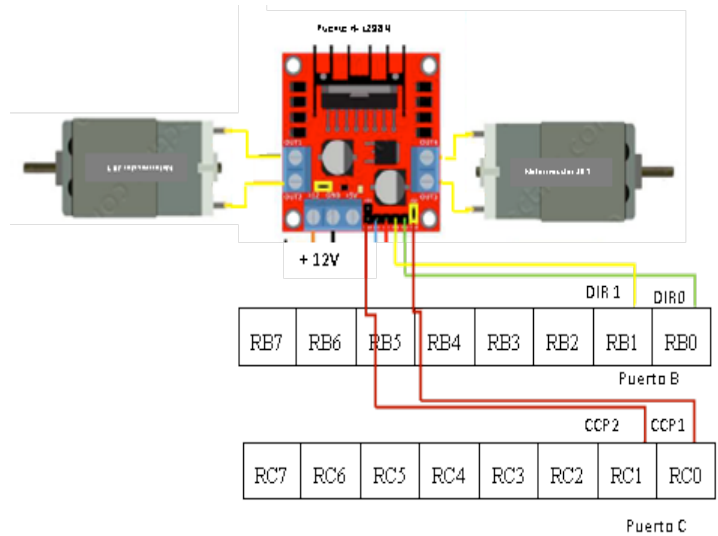


Figura 5. Diagrama esquemático del cableado del puente H con el puerto B y el puerto C.

B. Programación del Microcontrolador PIC18F4550

La programación del PIC se desarrolló en lenguaje c, donde se declaran las instrucciones del giro de los motores y la comunicación Bluetooth donde se envían las instrucciones de la aplicación del teléfono al PIC.

C. Aplicación desarrollada en APP Inventor y Ligthblue (LE)

La aplicación implementada en el proyecto se desarrolla bajo la plataforma de programación APP Inventor (para Android) y Lightblue (LE para iOS).

a) APP Inventor

Este entorno de desarrollo de software es creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. Donde el usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación.

Gracias a que es una interfaz amigable y de fácil uso que permite trabajar mediante una programación por bloques y posteriormente se genera un archivo. apk (figura 7), siendo este el archivo de instalación para el celular con sistema operativo Android.



Figura 6. Herramienta de Programación para desarrollo de aplicaciones para Android.

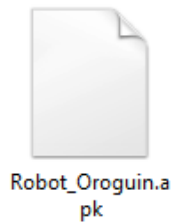


Figura 7. Archivo de instalación de la aplicación.

Una vez instalado se muestra el icono en la pantalla principal del celular y al abrir la aplicación aparece la interfaz del programa (figura 8).

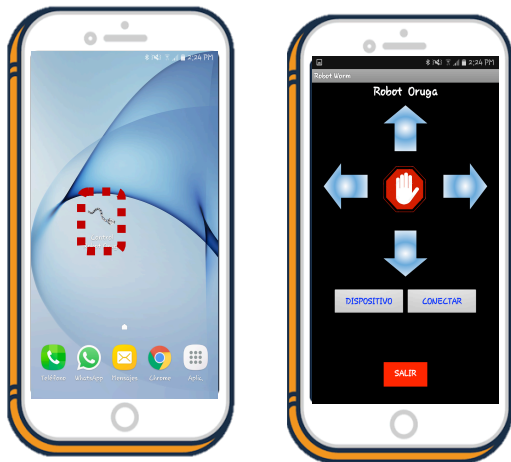


Figura 8. Aplicación móvil. (Imagen izquierda muestra el icono de la App y en la derecha la ejecución de la aplicación).

b) Aplicación en Ligthblue

Es una herramienta de programación que permite la conexión con todos los dispositivos que utilicen Bluetooth 4,0 de poca energía (también conocido como Bluetooth elegante, o luz de Bluetooth). Es una aplicación gratuita que se puede descargar directamente de APP Store. Tiene dos modos, centrales y periférico. En modo central, se puede explorar para conectar con todos los dispositivos de BLE alrededor de usted. Una vez que esté conectado, se deben tener en cuenta los detalles de los perfiles de todo el dispositivo, de los cuales se pueden leer y escribir las características y suscribir a las notificaciones. Esta herramienta es únicamente utilizada para dispositivos que tengan sistema operativo de iOS (figura 9).



Figura 9. Herramienta de Programación para desarrollo de aplicaciones para iOS.



Figura 9. Pantalla de la aplicación LightBlue al detectar uno o más bluetooths.

V. CONCLUSIÓN Y RESULTADOS

Se cumplen los objetivos propuestos y se logra una comunicación de forma eficaz entre el software y hardware mismos que permitieron la manipulación del robot con la aplicación que se desarrolló para el celular. De manera que el proyecto presentado, tiene la finalidad de brindar una herramienta que pueda ayudar a los rescatistas y las personas que requieran utilizar para la búsqueda, exploración o visualización de un espacio de difícil acceso para el humano.

Una de sus principales ventajas comparada con sistemas similares radica en su fácil implementación y además no requiere de hardware costoso para su utilización. Al igual que el tamaño compacto, bajo consumo de energía, facilidad de adquisición de componentes y bajo costo de mantenimiento.

La programación en lenguaje C favorece desarrollar diferentes programas gracias a su flexibilidad que permite trabajar con hardware de alta complejidad. Por lo que se logra una buena manipulación del robot tanto en la forma autónoma como en la teleoperada. Y la distancia alcanzada para el control del robot de forma teleoperada es suficiente para poder explorar zonas no accesibles.

Como trabajo futuro se planea incluir una cámara que permita la visualización del espacio a explorar.

Las imágenes siguientes muestran los resultados obtenidos.

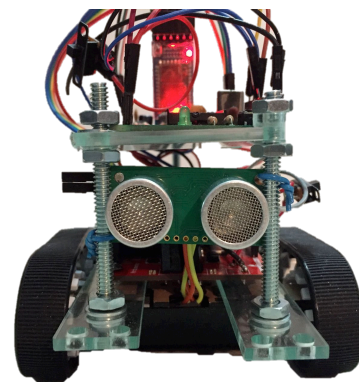


Figura 11. Robot tipo Oruga vista frontal.

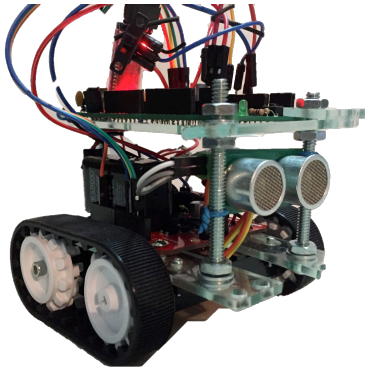


Figura 12. Robot tipo Oruga vista lateral.

REFERENCIAS

- [1] Bonet Esteban, E. V. (2017). *Lenguaje C*. [online] <http://informatica.uv.es/estguia/ATD/apuntes/laboratorio/Lenguaje-C.pdf> [Último Acceso 24 noviembre 2017]
- [2] Pololu.com. (2017). *Pololu - 6V High-Power (HP) Micro Metal Gearmotors*. [online] Available at: <https://www.pololu.com/category/173/6v-high-power-hp-micro-metal-garmotors> [Accessed 20 Nov. 2017].
- [3] De Las Nieves, V. (2011). *Control de motores de Corriente Continua-Puente en H*. Granábot.
- [4] Microchip Technology Inc. (2009) *PIC18F2455/2550/4455/4550*. Data Sheet. U. S. A.
- [5] HM Bluetooth module datasheet. *Bluetooth 4.0 BLE module. Datasheet V303*.
- [6] Tech Support. (2010) *HC-05, Bluetooth to Serial Port Module*.
- [7] Microchip Technology Inc. *PIC16F87XA Data Sheet*. (2003). 1st ed. U.S.A.
- [8] Núñez Olvera. O. F. Curso Avanzado de Microcontroladores (PIC18F4550). Instituto de Investigación en Comunicación Óptica . UASLP.
- [9] Jennifer Bray & Charles F. Sturman. *Bluetooth Connect Without Cables*, Second Edition , Prentice Hall PTR (2002)
- [10] Robert Morrow. *Bluetooth Operation and Use*, Mc Graw Hill (2002)

Control Servo Visual mediante OCR y Seguimiento de objetivos de un QuadRotor AR.Drone en LabVIEW

Juan Manuel Fortuna Cervantes.

Instituto de Investigación en Comunicación Óptica.
Universidad Autónoma de San Luis Potosí.
San Luis Potosí, México.
juan.manuel.fortuna@hotmail.com

Marco Tulio Ramirez Torres.

Coordinación Académica Región Altiplano Oeste.
Universidad Autónoma de San Luis Potosí.
Salinas de Hidalgo, S.L.P. México.
tulio.torres@uaslp.mx

Abstract— En este trabajo se presenta un sistema con el enfoque de control Servo Visual y la implementación de los algoritmos de visión por computadora, que hace el Reconocimiento Óptico de Caracteres (OCR en sus siglas en ingles) y el seguimiento de objetivos, de imágenes adquiridas por un vehículo aéreo no tripulado, como el Parrot AR.Drone 2.0. La primera aplicación permite al usuario enviar comandos al Drone a través de etiquetas de navegación; que son hojas con instrucciones impresas tales como Up=Subir, Down=Bajar, Left=Giro a Izquierda, Right=Giro a Derecha, Land=Aterrizar. La segunda aplicación considera la información del video, con un patrón a seguir en el plano de imagen de la cámara frontal, podemos alinear el Drone y llevar acabo desplazamientos verticales y laterales. Parte fundamental del sistema es que la computadora descifra el comando, utilizando un algoritmo OCR o Seguimiento respectivamente y envía al Drone la acción de vuelo. El uso del software LabVIEW junto con las herramienta de NI VISION, optimiza la toma de decisiones para el control de vuelo, además se hace uso de una estructura de programación paralela entre múltiples bucles lo cual evita la saturación de información en el sistema completo y permite la retroalimentación de acción de vuelo a través de un bucle de control. La conjunción metódica de estos procedimientos se ve aplicado a futuro para las tareas de vigilancia y la toma de decisiones, donde se emplean vehículos aéreos que realizan diferentes actividades de manera autónoma con altos niveles de rendimiento para aplicaciones en tiempo real.

Keywords—AR.Drone; LabVIEW; NI VISION; Control Servo Visual; OCR; Seguimiento de Objetivos.

I. INTRODUCCIÓN.

En los recientes años los QuadRotors se han convertido en las plataformas más utilizadas dentro del área de los vehículos aéreos no tripulados. En general su estructura es muy simple, pero existen algunos muy complejos con características especiales para la navegación. Un QuadRotor puede llegar a utilizar desde tres a doce rotores fijos para obtener un mejor desplazamiento o realizar diferentes maniobras [1], lo que lo hace una plataforma ideal para misiones de reconocimiento y seguimiento de objetivos.

Por otro lado, podría ser manipulado desde una estación de control en tierra con la intervención humana o de forma

autónoma en base a planes de vuelo preprogramados o con sistemas de automatización más complejos.

Una característica interesante de los QuadRotors es su capacidad para comunicar información valiosa, como la temperatura, imágenes o video de su entorno, así como, pueden disponer de los datos de estado, sobre la posición, la velocidad, dirección, altitud de vuelo y la carga de la batería. Los QuadRotors son muy utilizados, ya que pueden ejecutar un gran número de aplicaciones civiles y militares sin poner en riesgo vidas humanas. Además, pueden llevar a cabo maniobras rápidas y complejas [2], navegar de forma autónoma en entornos estructurados [3] y no estructurados [4]. En particular, se han aplicado sistemas de visión artificial, en la navegación [5-6], aterrizaje [7] y enjambres de QuadRotors [8-10]. La teoría de control, visión por computadora, mecánica, aerodinámica, automatización y la electrónica son algunas áreas relacionadas con el desarrollo de estos sistemas. Sin embargo, debido a los grandes avances en la tecnología, cada vez es necesario tener mejores algoritmos y mucho más eficientes para el control servo visual y así como el flujo de información de una manera más óptima y segura dentro de la estructura de programación. En relación a esto, la información depende en gran medida de la aplicación de ejecución y, en algunos casos, es necesario procesarla en tiempo real.

Este trabajo presenta una parte fundamental en el desarrollo de aplicaciones de reconocimiento y seguimiento de objetivos para los vehículos aéreos autónomos. Además, junto a la implementación de una estructura de programación que permite evitar la saturación de información entre operaciones de procesamiento de imágenes y el uso de las funciones de control de vuelo en múltiples bucles podemos tener un comportamiento estable del sistema. El resto de trabajo se organizará de la siguiente manera. La sección 2 presenta una breve revisión de la descripción del sistema. La sección 3 presenta una descripción general de la configuración experimental que se plantea. La sección 4 muestra la implementación de los algoritmos de reconocimiento y seguimiento de objetivos. Además, en la sección 5 podemos ver los resultados obtenidos. Finalmente, la sección 6 incluye las conclusiones.

II. DESCRIPCIÓN DEL SISTEMA.

A. Parrot QuadRotor AR.Drone 2.0.

El sistema principalmente se basa en el envío de datos a través de la comunicación Wi-Fi. El procesador del QuadRotor obtiene los datos adquiridos de los sensores y los procesa de acuerdo con la aplicación que ha ejecutado. Se toma en cuenta parte de la información generada por los ángulos de navegación y la posición en (X,Y) dentro del sistema (Figura 1), para la estabilización del mismo, además utiliza los datos del giroscopio y acelerómetro como referencia. Una vez que los datos han sido procesados, serán enviados a los controladores, quienes serán responsables de acelerar o disminuir la velocidad en cada par de rotores. Cada paquete de datos se procesará como en la siguiente Figura 2 [11-12].

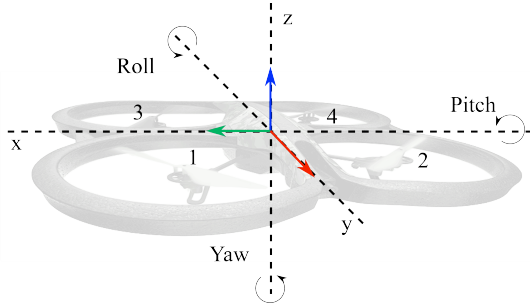


Figura 1. Orientación del QuadRotor definida por los ángulos Roll, Pitch e Yaw [13].

En nuestro sistema, LabVIEW se utiliza para obtener y manipular los datos generados por el QuadRotor. El proceso de comunicación y decodificación se realizan con la ayuda del Toolkit AR.Drone. Además, las imágenes tomadas por el QuadRotor se envían a la computadora portátil a través de las librerías dinámicas, que son funciones de LabVIEW. Una ventaja de utilizar el Toolkit AR.Drone es que los datos pueden extraerse, decodificarse, procesarse y nuevamente ser enviados al QuadRotor [14-15].

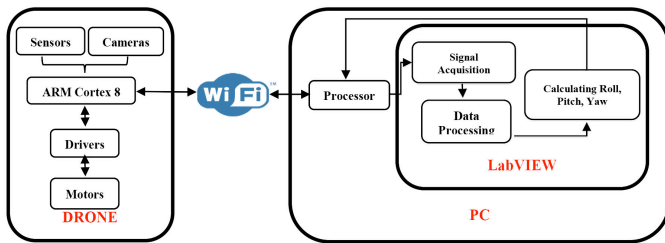


Figura 2. Diagrama de bloques del sistema.

B. Toolkit AR.Drone LabVIEW.

Parte fundamental de este Toolkit de desarrollo es que nos permite controlar el AR.Drone 1.0 y 2.0 con LabVIEW. Además, está integrado por VIs (*Virtual Instruments*) de alto nivel que realizan funciones importantes, como el control de navegación, lectura y decodificación de video y la recopilación de datos de los diferentes sensores. Estos VIs podrían usarse junto con las demás funciones incluidas en LabVIEW de manera gráfica o lenguaje gráfico de programación, para analizar los datos provenientes del AR.Drone y calcular los controles de salida que se enviarán de regreso.

Dentro de la paleta de funciones, *Open* es la función más importante respecto a esta investigación, el VI o función nos ayuda a abrir una nueva sesión con AR.Drone. Por otro lado, para obtener una transmisión de datos y establecer los parámetros de configuración, la PC debe estar conectada a la red inalámbrica generada por el AR.Drone. Además, podemos seleccionar el modo de vuelo, ya sea en modo interior o exterior, considerando así, el uso de la carcasa para interiores o prácticamente al aire libre. También, debemos tomar en cuenta que disponemos de dos cámaras de video, por lo que podemos seleccionar la que vamos a utilizar; una en la parte inferior y la otra en la parte frontal. Sin embargo, hay restricciones que se pueden configurar, como los valores máximos de navegación: altitud (m), ángulo de inclinación y desplazamiento (deg), velocidad vertical (m/s) y velocidad de navegación (deg/s).

La paleta de VIs se divide en varias funciones, otra función principal es *Control Drone*, la cual nos permite controlar el comportamiento de nuestro QuadRotor y, en otras palabras, es responsable de enviar los comandos básicos de control al AR.Drone, incluidos los comandos de movimiento (*Hover*, *TakeOff/Landing* y *Emergency Landing*). Además, tiene un grupo de cuatro elementos de entrada llamado *Control Cluster*, permitiendo manipular los parámetros de vuelo (*Roll*, *Pitch*, *Vertical Speed* y *Yaw Speed*). Dado a que es necesario poder visualizar lo que el QuadRotor está enviando a través de la comunicación Wi-Fi a la PC, usamos la función *Read NavData*. Cuenta con un grupo de salida de diez elementos; algunos datos importantes que podemos ver en los indicadores son: estado de control, porcentaje de batería y los parámetros de vuelo Yaw (deg), Pitch (deg) y Roll (deg). De la misma manera, podemos monitorear la altitud (cm) y la velocidad (cm/s) en varios ejes (v_x , v_y , v_z) después de haber inicializado la sesión.

Teniendo en cuenta algunas aplicaciones que requieren la parte de visión, es necesario leer los datos del video que se envían a través del puerto de red, con lo cual podemos usar la función *Read VideoData*. Cabe mencionar que los desarrolladores de National Instruments crearon una librería especial para la transmisión de los datos de video, cuyo nombre es *Decode20Video.dll*. Para toda estructura de programación y siguiendo el flujo de datos dentro de LabVIEW, es necesario que al final cualquier sesión abierta debe estar cerrada, usando otra función del Toolkit llamada *Close*.

C. Estructura de datos.

Los lenguajes de flujo de datos como LabVIEW permiten la paralelización de los bucles. El paralelismo es importante en los programas de PC porque pueden desbloquear el rendimiento en relación con los programas puramente secuenciales debido a los cambios recientes en los diseños de los procesadores [16]. Un caso particular, es la estructura Productor/Consumidor basada en el patrón Maestro/Esclavo y está orientado a compartir datos entre múltiples bucles [17]. En el que los bucles se ejecutan de forma independiente: el primer bucle es responsable de producir o recibir los datos y el segundo bucle solo se encarga de procesar los datos.

A pesar de correr en paralelo (diferentes velocidades), los datos se envían entre los dos bucles con la ayuda de las

funciones *Queue*, estructuras de datos estándar en lenguaje de programación. Principalmente se basa en la teoría primero en entrar/ primero en salir. Estas funciones se pueden inicializar fuera de los bucles Productor y Consumidor. La Figura 3 ilustra cómo se puede crear la estructura Productor/Consumidor en LabVIEW.

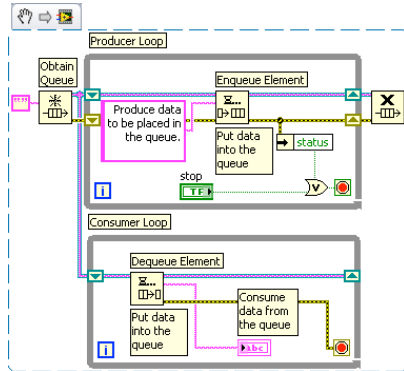


Figura 3. Estructura Productor/Consumidor se usa a menudo para aumentar el rendimiento de las aplicaciones que requieren tareas paralelas [17].

Este almacenamiento en memoria será muy importante cuando la comunicación de la red esté ocupada. Con el almacenamiento en búfer, los paquetes se pueden recuperar y comunicar más rápido de lo que se pueden analizar. Una parte fundamental y quizás una desventaja es no poder acceder aleatoriamente a los elementos dentro de los Queue, la única forma de ver y manipular todos los elementos que están dentro es tomarlos uno a uno a la vez.

III. CONFIGURACIÓN EXPERIMENTAL.

Para poder implementar la estructura de datos en paralelo junto con las funciones del Toolkit AR.Drone y además con las diferentes aplicaciones de control servo visual mediante el Reconocimiento Óptico de Caracteres y el seguimiento de objetivos y/o patrones, se consideró el lenguaje de programación gráfico LabVIEW, una marca registrada de National Instruments. La Figura 4 muestra el diagrama de bloques de la comunicación paralela, para las diferentes tareas que se ejecutarán en la aplicación. Se contempla el control manual de navegación desde el teclado de la PC, la lectura de la información de los sensores, los controles básicos de vuelo y la estructura Productor/Consumidor para la lectura de video, además con la implementación de los algoritmos propuestos.

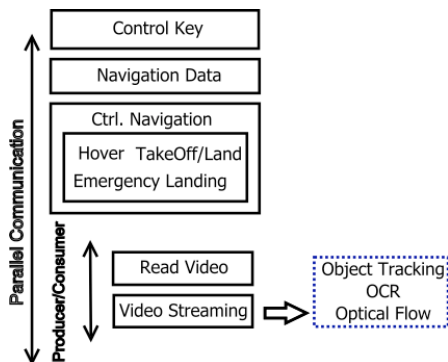


Figura 4. Diagrama de bloques de la comunicación paralela entre las diferentes tareas a realizar por AR.Drone y LabVIEW.

A. Reconocimiento Óptico de Caracteres.

Con las funciones de NI VISION para el procesamiento de imágenes en el video y el Toolkit AR.Drone, podemos integrar el siguiente diagrama de bloques para el control de vuelo (Figura 5), lo cual nos permite crear la tarea de control servo visual, tener la lectura de los caracteres y tomar las decisiones. Podemos utilizar las dos cámaras con las que cuenta el AR.Drone, pero para nuestra configuración solo utilizaremos la que está localizada en la parte frontal, dado que los comandos con etiquetas pueden ser colocados verticalmente con un ángulo de visión acorde para la cámara y así detectar la etiqueta de vuelo.

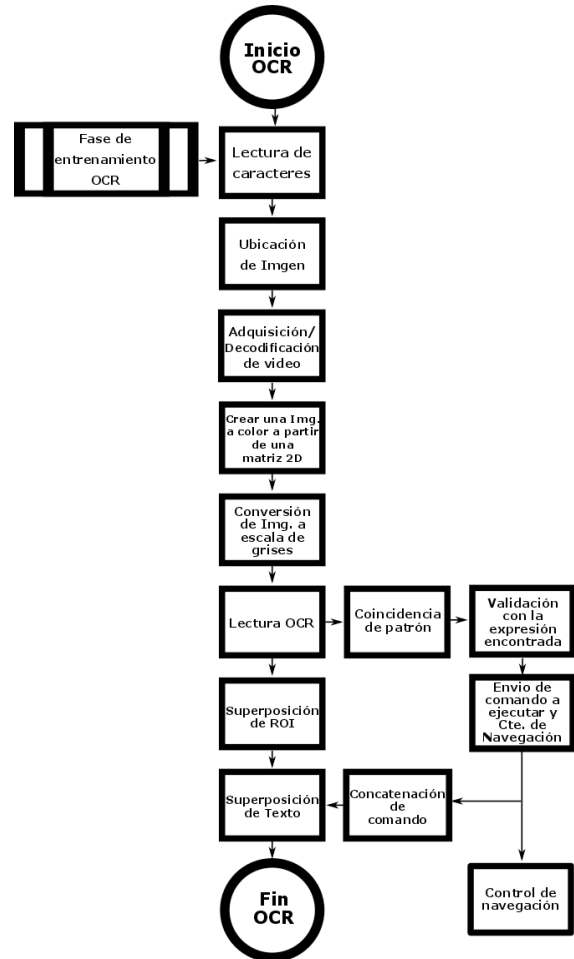


Figura 5. Diagrama de flujo para Control Servo Visual mediante Reconocimiento Óptico de Caracteres en LabVIEW.

B. Seguimiento de Objetivos.

Teniendo en cuenta las funciones de NI VISION y el Toolkit AR.Drone, podemos desarrollar la configuración experimental para la tarea de control servo visual (Figura 6). Con la búsqueda de patrones en el plano de imagen nos da pauta a integrar el algoritmo de seguimiento y alineación del AR.Drone. Cabe mencionar que dentro de la configuración inicial del AR.Drone solo utilizamos la cámara posicionada en la parte frontal, sin embargo, también es posible utilizar la cámara posicionada en la parte inferior de la estructura. También determinamos utilizar una ganancia constante de velocidad de 1 (m/s) para el algoritmo de navegación

Yaw/Vertical. La señal de error depende de las características en el plano de imagen, lo que se busca es generar las referencias de velocidad.

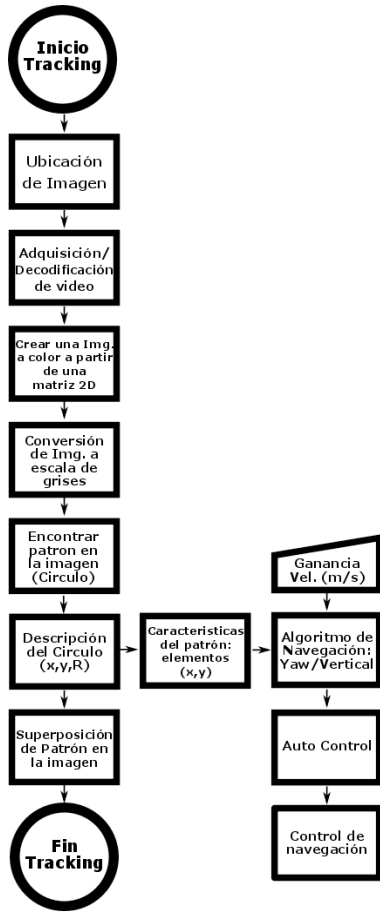


Figura 6. Diagrama de flujo para Control Servo Visual mediante Seguimiento de objetivos en LabVIEW.

IV. IMPLEMENTACIÓN DEL SISTEMA.

La estructura de programación en nuestro diagrama de bloques nos permite poder implementar nuestros algoritmos de control servo visual, de manera que podemos utilizar la comunicación paralela de múltiples tareas entre el AR.Drone y LabVIEW. Principalmente partimos de la adquisición y decodificación de imágenes o video desde otra etapa donde encontramos parte de la estructura Productor/Consumidor, en este caso el bucle Productor, en donde usamos la función de *Read VideoData*. A continuación describimos la etapa de implementación de los algoritmos de forma independiente.

A. Reconocimiento Óptico de caracteres.

El algoritmo para el control servo visual, está implementado en varias etapas, como podemos ver en la Figura 7, en el bucle con nombre Consumidor se va creando una localidad temporal de memoria de la información enviada, está a su vez sufre una conversión a escala de grises, donde pasara a la etapa de lectura de caracteres. En la Figura 8 podemos ver la etapa de validación a través de la coincidencia de patrones de las etiquetas, con esto se logra enviar los comandos de navegación a ejecutar y se envía la constante de vuelo al control de navegación.

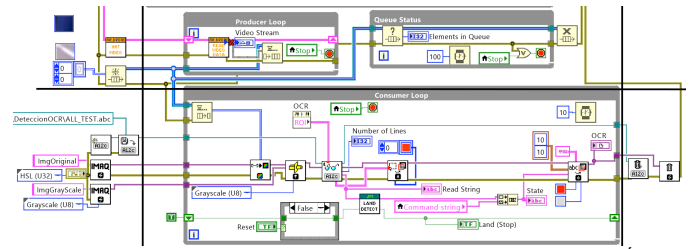


Figura 7. Algoritmo Control Servo Visual mediante Reconocimiento Óptico de Caracteres implementado en el Diagrama de bloques de LabVIEW.

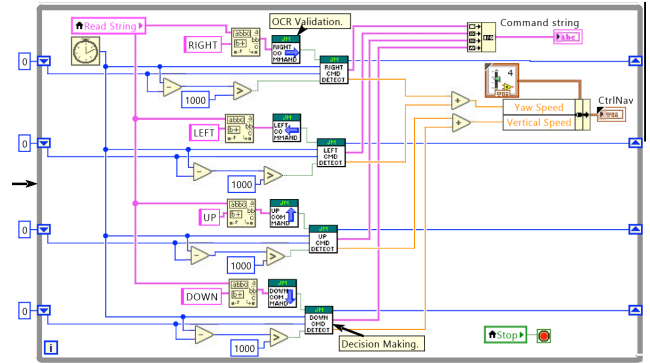


Figura 8. Etapa de validación y toma de decisiones para el Control Servo Visual mediante Reconocimiento Óptico de Caracteres.

B. Seguimiento de Objetos.

Implementando el algoritmo para el control servo visual, consideramos también algunas etapas durante la aplicación, como podemos ver en la Figura 9, en el bucle principal de la estructura Productor/Consumidor, apilamos toda la información enviada por video, para después ser enviada a la estructura Consumidor. Dentro de este bucle se genera una localidad de memoria de la información enviada, para después aplicar un procesamiento en la imagen, una conversión a escala de grises. Siguiendo el diagrama de flujo de la configuración experimental, debemos encontrar un patrón en la imagen (Circulo), al detectar el patrón debemos obtener la información o descripción del circulo como la posición (x, y) y el radio. A partir de las características del patrón (x, y), podemos utilizar esta información para implementar nuestro algoritmo de navegación Yaw/Vertical junto a una ganancia de velocidad (m/s) determinada para el sistema. De manera general, para nuestro algoritmo, utilizamos la información del plano de imagen (centro del círculo), con el cual es utilizado para alinear y seguir la trayectoria y posición del AR.Drone.

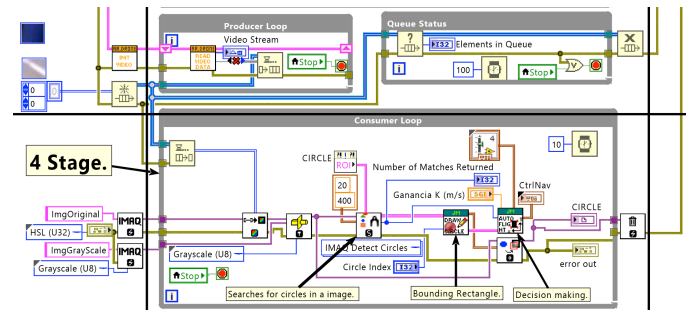


Figura 9. Algoritmo Control Servo Visual mediante el seguimiento de objetivos implementado en el diagrama de bloques de LabVIEW.

V. RESULTADOS.

Una forma de presentar los resultados experimentales obtenidos en tiempo real mediante la implementación de la estructura Productor/Consumidor, monitoreamos de manera gráfica los datos de navegación y de manera paralela la transmisión de video para las diferentes algoritmos de control servo visual. En particular, se puede ver en Figura 10 (a-c), los resultados de la prueba experimental antes de ejecutar las aplicaciones de control servo visual, un comportamiento oscilatorio en la fase de los ángulos de navegación. Una parte importante de estos resultados es el tiempo de ejecución en la adquisición de datos, dado a que es continuo hasta que el mismo usuario detiene la ejecución de la aplicación sin tener problemas durante el proceso.

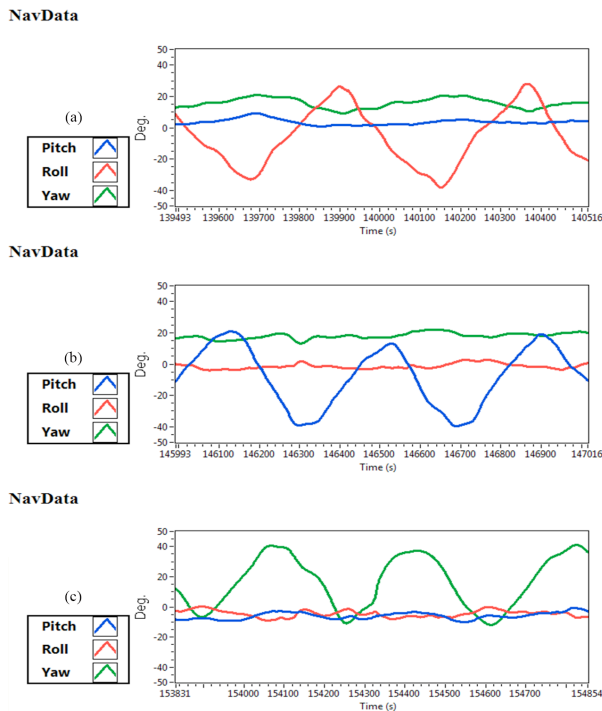


Figura 10. Resultado experimental sin tener en vuelo el AR.Drone en donde se visualiza una oscilación en (a) Roll, (b) Pitch y (c) Yaw.

Como resultado de esta implementación y a partir de la transmisión de video, dentro del espacio de trabajo podemos encontrar las diferentes acciones de navegación con la ejecución de los algoritmos de control servo visual. En la Figura 11 podemos visualizar las acciones de control referente al control servo visual mediante el Reconocimiento Óptico de Caracteres. Además, en la Figura 12 visualizamos las acciones de control mediante el seguimiento de patrones.

Resulta indispensable la integración de referencias visuales y máquinas de estado para controlar la navegación autónoma en diferentes estados de vuelo, así como los desplazamientos verticales, horizontales y longitudinales del QuadRotor. Las aplicaciones han sido desarrolladas, analizadas y validadas usando LabVIEW, a través de algunas pruebas experimentales las cuales consideran factores como velocidad de procesamiento, iluminación, vibraciones, y condiciones ambientales que influyen en el rendimiento del sistema.



Figura 11. Control Servo Visual mediante el Reconocimiento Óptico de caracteres, etiquetas con comandos de control de navegación propuestos (a) Down, (b) Up, (c) Right, (d) Left, (e) Land y (f) Modo Land.



Figura 12. Control Servo Visual mediante Seguimiento de objetivos (Tracking), modos de vuelo (a) alineación de QuadRotor AR.Drone con objetivo (centro del círculo), (b-c) control de navegación Yaw y (c) control de navegación vertical.

VI. CONCLUSIONES.

En este trabajo, presentamos AR.Drone como una plataforma utilizable en la investigación de la robótica y el control servo visual. El desarrollo y uso de algoritmos de procesamiento visual más fiables y robustos, permite que el control visual se lleve a cabo sin la necesidad de fusionar la información inercial del QuadRotor. Además, no es solo en el hardware del QuadRotor donde hay avances tecnológicos, sino también la parte de software donde se logra la comunicación entre los sistemas integrados con la implementación de las estructuras de datos (Productor/Consumidor) dentro del sistema en LabVIEW y los algoritmos de visión. Con el almacenamiento de la información, los resultados obtenidos proporcionan un buen flujo de datos, mejorando la tasa de transmisión y una gran flexibilidad para enviar la señal de video en tiempo real, además, sin tener problemas en la saturación en

cada función implementada en el diagrama de bloques. De esta forma, podría ser una estructura base para implementar otros algoritmos de visión artificial que permitan el desarrollo de aplicaciones como navegación autónoma, toma de decisiones, reconstrucción digital aérea y la búsqueda en entornos donde el acceso es complicado.

AGRADECIMIENTOS.

Expreso mi agradecimiento al Instituto de Investigación en Comunicación Óptica y al Consejo Nacional de Ciencia y Tecnología que a través del programa nacional de posgrados de calidad se me permitió realizar este trabajo de investigación.

REFERENCIAS.

- [1] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Trans. Neural Networks*, vol. 21, no. 1, pp. 50–66, 2010.
- [2] Mellinger, D., Michael, N., Kumar, V. "Trajectory generation and control for precise aggressive maneuvers with quadrotors". International Symposium on Experimental Robotics, Delhi, India, 2010.
- [3] Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N. "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments". SPIE Unmanned Systems Technology XI, Orlando, F, vol. 7332, 2009.
- [4] Blöndsch, M., Weiss, S., Scaramuzza, D., Siegwart, R. "Vision based MAV navigation in unknown and unstructured environments". IEEE Int. Conf. on Robotics and Automation, pp. 21–28, 2010.
- [5] B. Odile, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Kinematic Visual Servo Control of a Quadrotor Aerial Vehicle", vol. 25, no. 3, pp. 833–838, 2007.
- [6] W. Zheng, F. Zhou, and Z. Wang, "Robust and accurate monocular visual navigation combining IMU for a quadrotor," *IEEE/CAA J. Autom. Sin.*, vol. 2, no. 1, pp. 33–44, 2015.
- [7] J. Park, Y. Kim, and S. Kim, "Landing Site Searching and Selection Algorithm Development Using Vision System and Its Application to Quadrotor," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 488–503, 2015.
- [8] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks", *Proc. IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.
- [9] Y. Wang, Q. Wu, and Y. Wang, "Distributed consensus protocols for coordinated control of multiple quadrotors under a directed topology", *IET Control Theory Appl.*, vol. 7, no. 14, pp. 1780–1792, 2013.
- [10] D. Lee, A. Franchi, H. Il Son, C. Ha, H. H. Bulthoff, and P. R. Giordano, "Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles", *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.
- [11] R. Lozano, P. Castillo and A. E. Dzul. "Modelling and control of miniflying machines", Springer, Londres, 2005.
- [12] G. Tont, R. Munteanu, D. Tont, D. Iudean, "Aspects Regarding the Unidirectional Two-Port Circuits Implemented by Means of Electronic Gytrators", *Journal of Advances in Electrical and Computer Engineering*, Vol. 16, no. 1, 2016.
- [13] J. Engel, "Autonomous Camera-Based Navigation of Quadrocopter", Master's Tesis in Informatics, Technical University of Munich, December 15, 2011.
- [14] "AR.Drone Toolkit for LabVIEW". Jun. 2017, National Instruments. sine.ni.com/nips/cds/view/p/lang/es/nid/211837
- [15] R. Munteanu, D. Iudean, C. Muresan, V. Petean, "Acess System in Restricted áreas base on programable logic controller and GSM modern", The 13th International Conference on Engineering of Modern Electric Systems (EMES), 2015.
- [16] "Benefits of Programming Graphically in NI LabVIEW". Jun. 2013, National Instruments. www.ni.com/white-paper/14556/en/.
- [17] "Application Design Patterns: Producer/Consumer". Aug. 2016, National Instruments. www.ni.com/white-paper/3023/en/.

Compresión de señales usando transformadas ortogonales

Reyes-López L.E. & Castro García R.
Instituto de Investigación en Comunicación Óptica (IICO)

Resumen—En la actualidad han surgido un gran número de aplicaciones para el procesamiento de señales. Una de las principales características en dichas aplicaciones es la necesidad de una gran cantidad de recursos para el almacenamiento y transmisión de tales señales. Este análisis se realiza por medio de dos herramientas matemáticas: la transformada de Fourier y la transformada de Haar, ya que estas surgen como herramientas poderosas para procesar de manera eficiente las señales que involucran grandes cantidades de información.

I. INTRODUCCIÓN

En matemáticas, la wavelet de Haar es una cierta secuencia de funciones. En la actualidad se le reconoce como la primer wavelet conocida. Esta secuencia fue propuesta en 1909 por Alfred Haar. Haar usó estas funciones para dar un ejemplo de un sistema ortonormal contable para el espacio de las funciones de cuadrado integrable en la recta real. El estudio de las wavelets, e incluso el término "wavelet", no vinieron hasta mucho después. Como un caso especial de wavelet de Daubechies, también es llamado *db2*. La wavelet de Haar es también la wavelet más simple posible.

La desventaja técnica de la wavelet de Haar es que no es continuo y por lo tanto no derivable. Esta propiedad, de cualquier forma, es una ventaja para el análisis de señales con transiciones repentinas, tales como el monitoreo del fallo de una herramienta en una máquina.

II. MARCO TEÓRICO

El uso de una transformada matemática tiene como uno de sus principales objetivos encontrar una representación alternativa en un dominio diferente que muestre características que en el dominio original serían difíciles o casi imposibles de detectar. Además, en muchas situaciones con las propiedades de la transformación pueden facilitar o realizar de manera más flexible operaciones de carácter analítico o numérico.

II-A. Transformada de Haar

En el caso de la transformada de Haar (TH), como se observará, tendremos la descomposición de una señal en términos tanto del tiempo y en el dominio de la escala mediante los procesos de dilatación y traslación. En este trabajo se considera la versión discreta de la TH, la cual

considera un conjunto de funciones wavelet que conforman una base de carácter ortogonal (u ortonormal) en $L^2(\mathfrak{R})$. En este caso, la TH se lleva a cabo en una versión discreta de los parámetros de dilatación y traslación a y b , respectivamente. Dentro de este contexto, una señal arbitraria $x(t)$ de energía finita se puede expresar en términos de una base ortonormal wavelet como:

$$x(t) = \sum_m \sum_n d_{m,n} \psi_{m,n}(t)$$

donde los coeficientes de la expansión se dan por la expresión

$$d_{m,n} = \int_{-\infty}^{\infty} x(t) \psi_{m,n}(t) dt.$$

II-B. Transformada de Fourier

En matemáticas, la transformada discreta de Fourier (TFD), es un tipo de transformada discreta utilizada en el análisis de Fourier. La TFD requiere que la función de entrada sea una secuencia discreta y de duración finita, dichas secuencias se generan a partir del muestreo de una función continua, como puede ser la voz humana. La entrada de la TFD es una secuencia finita de números reales o complejos, de modo que es ideal para procesar información almacenada en soportes digitales. En particular, la TFD se utiliza comúnmente en procesamiento de señales y otros campos relacionados dedicados a analizar las frecuencias que contiene una señal muestreada para resolver ecuaciones diferenciales parciales, y para llevar a cabo operaciones como convoluciones o multiplicaciones de grandes números enteros. Un factor muy importante para este tipo de aplicaciones es que la TFD puede ser calculada de forma eficiente en la práctica utilizando el algoritmo de la transformada rápida de Fourier o FFT (Fast Fourier Transform).

Para iniciar el análisis la frecuencia ω se hace discreta. La señal $x[n]$ es una secuencia de N puntos. Entonces la TFD podemos calcular de la siguiente forma

$$X(\omega) = \sum_n x[n] e^{-j\omega n}.$$

En cuestión de la energía se cumple el mismo principio, demostrado a continuación

$$E_x = \sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2$$

La energía se puede calcular, ya sea con la secuencia o con la señal transformada, debido a la relación de Parseval, ya que son equivalentes.

III. DESARROLLO

Se emplea el esquema de compresión implementado en Matlab mostrado en la Figura 1.

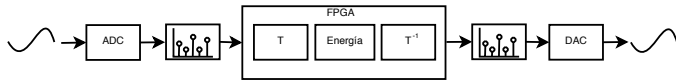


Fig. 1. Esquema de compresión propuesto.

Se propone un esquema de compresión para la aplicación real del esquema de compresión compuesto por un ADC, FPGA y un DAC para la implementación del mismo. Con este esquema se busca adecuar la señal de audio con la ayuda de Matlab, y así calcular la transformada de Fourier y la transformada de Haar, para calcular y obtener su mapa de energía acumulativa, posteriormente comprimir las señales y calcular sus transformaciones inversas respectivamente y así obtener los resultados. Para ello, se consideran señales de audio con una longitud de 16384 datos, se muestran en la sección IV.

IV. RESULTADOS

En esta sección se muestran diferentes tasas de compresión las cuales se aplicaron al esquema propuesto,

- a) Señal de audio cargada en Matlab, estos son las gráficas considerando un umbral de 60%:

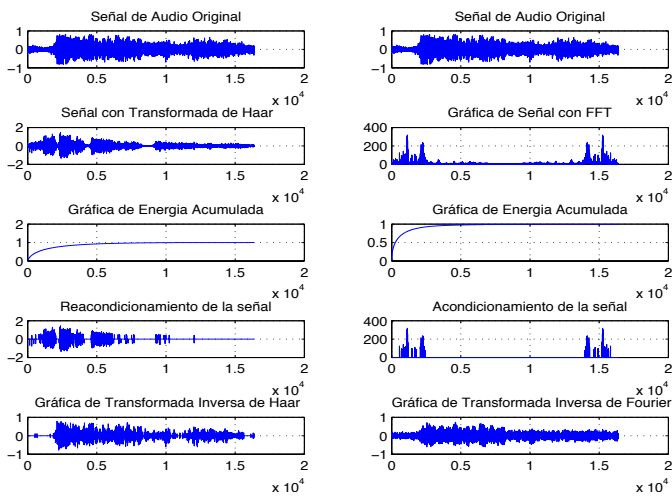


Fig. 2. Gráfica de resultados considerando un 60% de umbral.

- b) Señal de audio cargada en Matlab, estos son las gráficas considerando un umbral de 70%:

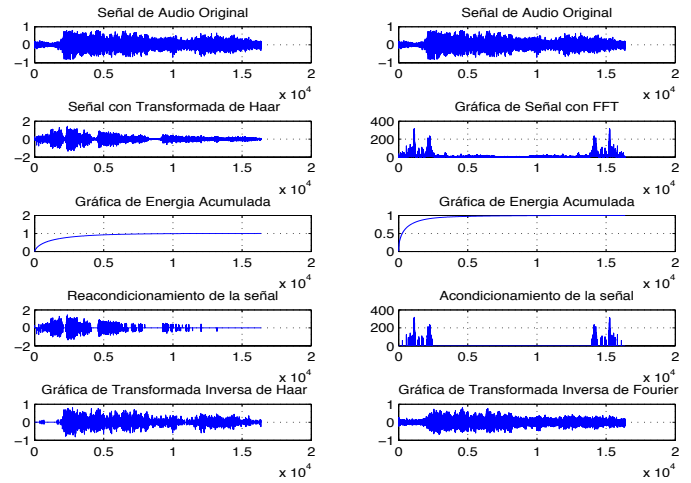


Fig. 3. Gráfica de resultados considerando un 70% de umbral.

- c) Señal de audio cargada en Matlab, estos son las gráficas considerando un umbral de 95%:

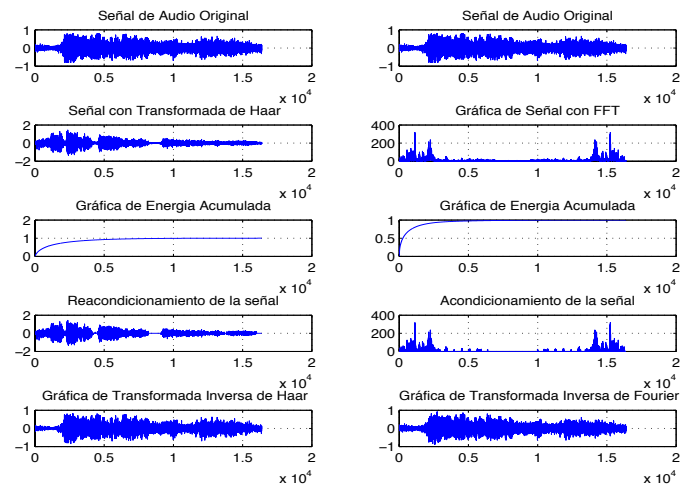


Fig. 4. Gráfica de resultados considerando un 95% de umbral.

Como se puede ver en la Figura 2, se está considerando un porcentaje de umbral de 60% que corresponde a un valor de umbral de 0.3842 usando la transformada de Haar y un valor de umbral de 72.2319 en el caso de la transformada de Fourier, es decir, que los valores que estén por debajo de estos valores se tomaran como cero, a partir de ahí se tomaran los valores que quedan debido a la discriminación de datos, sobre los valores que quedan se realiza el cálculo de la transformada inversa correspondiente. En este caso se puede apreciar en la gráfica se tiene un gran número de valores que se toman con el valor de cero.

En la Figura 3, se toma un valor diferente de umbral, y se aprecia claramente que los valores que quedan después de la discriminación son mejores, y estos resultados se muestran en las Tablas 1 y 2. En este caso se tiene un umbral de 70% en ambos casos se tiene una tasa de compresión de 1:9 usando la transformada de Haar, y en el caso de la transformada de

Fourier se tiene una tasa de 1:23. En este caso en particular, la transformada de Fourier tiene mejores resultados, que la transformada de Haar.

En lo que respecta a la Figura 4, es el caso donde el porcentaje de umbral es de 95%, es decir, la compresión debería de ser mínima, el valor correspondiente a este porcentaje es de 0.1162 en el caso de la transformada de Haar, el valor de umbral con la transformada de Fourier es de 15.9782. Sin embargo, se tiene una compresión significativa, se tuvo una tasa de compresión de 1:2 y de 1:5 con la transformada de Haar y la de Fourier, respectivamente.

En la tabla se tiene los resultados experimentales considerando diferentes porcentajes de umbral con la transformada de Haar.

% de umbral	Valor de umbral	Números de valores	Tasa de compresión
60%	0.3842	1133	1:14
70%	0.3112	1697	1:9
80%	0.2407	2588	1:6
90%	0.1633	4294	1:3
95%	0.1162	6052	1:2

Tabla 1. Resultados experimentales usando la transformada de Haar a diferentes porcentajes de umbral.

En la tabla se tiene los resultados experimentales considerando diferentes porcentajes de umbral con la transformada de Fourier.

% de umbral	Valor de umbral	Números de valores	Tasa de compresión
60%	72.2319	434	1:37
70%	57.8642	696	1:23
80%	43.2706	1134	1:14
90%	27.3383	2047	1:8
95%	15.9782	3269	1:5

Tabla 2. Resultados experimentales usando la transformada de Fourier a diferentes porcentajes de umbral.

En las Tablas 1 y 2, se muestran los valores experimentales obtenidos al emplear el esquema de compresión

V. CONCLUSIONES

La aplicación de transformadas en cualquier tipo de señales nos facilita los cálculos matemáticos ya que nos dan su respuesta en frecuencia, y es más sencillo realizar cálculos usando transformadas. Se observó que esta aplicación ofrece una aplicación real del uso de transformadas, y como pueden facilitar el uso de actividades que actualmente se desarrollan con normalidad; como lo es la transmisión de datos, este ha sido una aplicación donde ha tenido gran impacto el uso de softwares que compriman datos, pero estos aún no son optimizados, o no pueden comprimir a la tasa que el usuario requiera, debido a esto, este análisis que se realiza tiene gran significado para poder llevarlo a la práctica. Se vio en los resultados se tiene una buena tasa de compresión a diferentes porcentajes de umbral, sin embargo, existe un compromiso que se tiene entre tasa de compresión y la calidad de señal que

se recupera, debido a esto se tiene una buena tasa de compresión y una buena calidad de la señal que se recupera entre 70% y 85%.

REFERENCIAS

- [1] Khalifa, O. O., Harding, S. H., & Hashim, A. H. A. (2008). Compresion using wavelet transform. *Signal Processing: An International Journal*, 2(5), 17-26.
- [2] Reddy, B. S., & Murthy, I. S. N. (1986). ECG data compression using Fourier descriptors. *IEEE Transactions on Biomedical Engineering*, (4), 428-434.
- [3] Daubechies, I. (1992). Ten lectures on wavelets (Vol. 61). Siam.
- [4] Akin, M. (2002). Comparison of wavelet transform and FFT methods in the analysis of EEG signals. *Journal of medical systems*, 26(3), 241-247.

Robot móvil con una trayectoria basada en un sistema caótico de Lorenz

Domínguez-Rivera I.^a, Villa Salas L.R.^b, L.J. Ontañón García^c

Coordinación Académica Región Altiplano Oeste,
Universidad Autónoma de San Luis Potosí,
Carretera Salina-Santo Domingo, #200, C.P. 78600
Salinas de Hidalgo, San Luis Potosí, México

^aisac004_@hotmail.com

^bluis_villa_salas@hotmail.com

^cluis.ontanon@uaslp.mx

Resumen— El objetivo principal es interpretar una función de un sistema caótico con un robot físico, pensado en la forma que tomará al utilizar un plano del sistema caótico para poder ver una representación 2D del mismo y que a su vez sea fácil que el robot pueda interpretar dicha función, en este caso ya que es una parte de un sistema caótico hace impredecible el movimiento del robot, con lo que se vio que es demasiado útil para utilizarlo como sistema de seguridad para que este no pueda ser predicho.

I. INTRODUCCIÓN

Con un poco de matemáticas básicas y algunos toques de programación echar a andar un robot ciertamente no es difícil, pero hacer que este siga una línea del exterior cambia la dificultad, ahora pensar en que este siga una línea de una función o algo parecido suena un poco más interesante.

Para este proyecto se creó un algoritmo dentro de MATLAB enlazado con un LEGO Mindstorm que es el robot que interpretará la función, primero es necesario que el robot conozca “la línea” por la que este se va a desplazar, el algoritmo trabaja con puntos matriciales por lo que se tiene que colocar en un punto inicial o punto de partida desde el que el robot se desplazará, el algoritmo calcula la distancia entre dos puntos como se muestra en la siguiente fórmula:

$$\overline{AB} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad [1]$$

Para calcular el ángulo entre dos puntos se utiliza la siguiente fórmula:

$$\phi = \tan^{-1} \frac{y_2 - y_1}{x_2 - x_1} \quad [2]$$

Con ambas ecuaciones como se muestra en la Figura 3 el robot podrá avanzar y girar respectivamente para seguir la forma de la función que se le da.

II. PROGRAMA

Para poder echarlo a andar fueron necesarios conocimientos básicos de MATLAB que fue el programa con el que se realizó la conexión con el robot LEGO Mindstorm y geometría básica, se le cargaba al robot el programa con el que el brick en base a calcular el ángulo entre tres puntos conocía si girar hacia la derecha o hacia la izquierda dependiendo del ángulo, una vez esto hecho el robot comenzaba a girar hasta que la lectura del giroscopio fuera igual a la calculada, una vez encontrado el ángulo empezaba el movimiento hacia el siguiente punto.

Para poder manejar enviarle el código a el robot era necesario antes inicializar los motores y el robot (como se muestra en la figura 1) con una asignación de nombre para que el programa reconociera el brick y de esta manera también los comandos para poder darle las instrucciones desde el programa.

Con el código de la figura 2 el robot calculaba el ángulo que necesitaba girar, se chequeaba si el ángulo era positivo o negativo y cuanta era la diferencia entre este y el ángulo anterior para buscar giro más pequeño hacia el la dirección del siguiente punto; si diferencia entre los ángulos era mayor a el ángulo actual se le sumaban 360 grados para que este fuera en la misma dirección, claro también revisando si el ángulo actual era positivo o negativo.

En la figura 3 se muestra el código para calcular la distancia entre los puntos y esto se iba guardando en un arreglo, también se modifica el ángulo suponiendo que delante es la derecha, si el punto llegase a estar hacia atrás o bien la izquierda a el ángulo se le agregaban 180 grados para que girara hacia la izquierda debido a que el ángulo se calculaba con respecto a el eje de las x por lo que este estaba en la dirección contraria respecto a la configuración del giroscopio.

Lo ideal sería agregarle más sensores para que el robot pueda esquivar obstáculos, a su vez conocer el tamaño o forma de el obstáculo para recalculer la trayectoria hacia el punto destino.

III. MATERIALES Y MÉTODOS

- Brick programable Lego mindstorm EV3
- Sensor giroscopio
- 2 servomotores 360°
- 2 ruedas
- MATLAB

Primeramente para poder mover el robot basándose en medidas reales se hizo un escalamiento midiendo el diámetro de las ruedas utilizadas para saber cuántos grados debía de girar para avanzar un centímetro, después a partir de la matriz de puntos se calculaba la distancia utilizando de 2 en 2 puntos y se guarda en otra matriz, esta se multiplicaba por un escalar que se puede modificar dependiendo de qué tan grande queramos que sea el área a cubrir.

También a partir de los puntos calculados se medía el ángulo que existe entre cada 3 puntos, esto con el fin de saber hacia dónde debe de girar y en qué sentido para no hacer un giro innecesario y perder tiempo. Ambas tareas fueron realizadas aplicando formulas básicas del algebra que serían la de calcular la distancia entre 2 puntos en el plano cartesiano y la del ángulo entre 2 rectas, y estas mediciones iban siendo realizadas durante la ejecución del programa para después realizar los movimientos correspondientes.

IV. RESULTADOS Y DISCUSIÓN

El proyecto tuvo resultados regulares y buenos a la hora de hacer lo que debía de hacer, algunos problemas que se presentaban eran que una de las dos llantas algunas veces no tenía buena tracción, esto provocaba que se desviara de la trayectoria original al robot y cambiara el ángulo final del movimiento, otro fue que el ángulo medido por el giroscopio no era exacto al calculado por las fórmulas ya que no aceptaba decimales por lo tanto se tenía que redondear y otro problema fue que a veces al girar para llegar al ángulo deseado este se pasaba, de ahí en más se piensa que se tuvieron los buenos resultados, ya que el movimiento realizado por el robot era bastante similar al calculado en las fórmulas, esto al ser la primera vez de experimentar algo así y probar las características de estos robots se sintió que fue buena la respuesta de los métodos y lógica utilizados. Al ver que funciono se podría implementar con materiales de más precisión y hacer algo mejor con una cierta aplicación como la de seguridad.

Es posible continuar con el proyecto ya que lo ideal sería agregar más funciones como el esquivar obstáculos pudiendo

el robot recalculer la trayectoria que necesita seguir para evitar la colisión con dicho objeto, también se pensó implementar un método para que en base a una imagen y un camino sobre esta misma el robot siga dicha línea.

V. CONCLUSIÓN

Este fue un proyecto que dejo buenos y mejores resultados a los especulados ya que se observó que podía graficar una función basándose en los puntos obtenidos sustituyendo por un valor las incógnitas de esta, esta graficación era bastante similar a la de la graficación en algún programa como GeoGebra, razón por la cual se podrían aplicar los métodos utilizados en alguna impresora 2D u otra aplicación similar como la de láseres para dibujar o CNCs. Como se ha mencionado hubo ciertos problemas los cuales se piensa que podrán ser solucionados con el uso de materiales de mayor calidad o la implementación de algoritmos para reducir el error físico y hacerlo más asertivo.

```
clc
clear
Walle = legoev3('USB');
%% Declaracion motores
motorIZQ = motor(Walle,'D');
motorDER = motor(Walle,'A');
%% Declaracion sensores gyro
giros = gyroSensor (Walle,1);
```

Figura 1: Código de la inicialización de los motores

```
while m < j
    if P(m,1)>P(m+1,1)
        if Ang(m) >= 0
            ang2 = abs(Ang(m)) + (Ang(m-1));
            ang3 = Ang(m) + 360 - Ang(m-1);
            switch ang2 && ang3
                case ang2 > ang3
                    Ang2(m) = Ang(m)+360;
                case ang2 < ang3
                    Ang2(m) = Ang(m);
            end
        elseif Ang(m) < 0
            ang2 = abs(Ang(m)) - (Ang(m-1));
            ang3 = Ang(m) + 360 + Ang(m-1);
            switch ang2 && ang3
                case ang2 > ang3
                    Ang2(m) = Ang(m)+360;
                case ang2 < ang3
                    Ang2(m) = Ang(m);
            end
        end
    else
        Ang2(m) = Ang(m);
    end
    m = m+1;
end
```

Figura 2: Código del cálculo del ángulo


```

while i < j-1

    r = sqrt((P(m+1,1)-P(m,1))^2+(P(m+1,2)-P(m,2))^2); % Es la formula de la distancia entre dos
    puntos sqrt((x2-x1)^2+(y2-y1)^2)
    if P(m,1)>P(m+1,1)
        ang = atand((P(m+1,2)-P(m,2))/(P(m+1,1)-P(m,1))) - 180; % Es la misma formula de abajo
        pero se le restan 180 grados debido a que va hacia atras suponiendo que adelante es hacia la
        derecha
    else
        ang = atand((P(m+1,2)-P(m,2))/(P(m+1,1)-P(m,1))); % Se representa la formula del angulo
        entre dos puntos en el plano (tan^-1(y2-y1)/(x2-x1))
    end

    u = round(ang); %round redondea el valor hacia el valor entero mas cercano
    x = r * 3; % Se realiza la multiplicacion de los centimetros por la escala que en este caso
    es 5 cm

    Mov(m) = x;
    Ang(m) = u;

    i = i+1;
    m = m+1;
    pause(0.1)
end

```

Figura 3: Código del cálculo de la distancia entre los puntos

References

- [1] D. Zaldívar Navarro, E.V. Cuevas Jiménez, M.A. Pérez Cisneros. Proyectos con robots LEGO. Bogotá, Colombia. Ra-Ma Ediciones de la U. 2015.pp. 205-207.

Rutinas específicas programadas con máquinas de estados en LabVIEW para AR.DRONE 2.0

Martínez López Maximiliano¹, Ramírez Torres Marco Tulio¹.

¹Coordinación Académica Región Altiplano Oeste, UASLP, Salinas, S.L.P.
maxmodo@live.com.mx

Resumen— En este proyecto de investigación como trabajo de tesis, se presenta la programación de rutinas específicas para el vehículo aéreo no tripulado Ar.Drone 2.0 de Parrot, utilizando el lenguaje de programación gráfico LabVIEW. Con dichos algoritmos el drone realiza rutinas de manera automática, es decir, se desplaza en sus diversos grados de libertad (Roll, Pitch, Vertical, Yaw), recorre ciertas áreas específicas, toma fotos y videos de su entorno utilizando las cámaras integradas. Estas rutinas fueron posibles gracias al toolkit especial AR.Drone el cual nos permitió acceder al vehículo no tripulado y poder tomar lectura de los sensores, controlar los actuadores, adquirir imágenes y video, así como combinar estas instrucciones con las estructuras y módulos de LabVIEW como ciclos, módulo de visión (NI-Vision), tiempos específicos, condicionales, etc. El diseño de máquinas de estado nos permite dividir cada tarea desarrollada en diferentes pasos o etapas. Los drones en la actualidad tiene una amplia gama de aplicaciones, por ejemplo se utilizan en la búsqueda de personas, monitorización de campos agrícolas, entre otra más. El objetivo de este proyecto es que el drone pueda realizar nuevas tareas de manera programada acorde a las necesidades del usuario.

I. INTRODUCCIÓN

Los drones son vehículos aéreos no tripulados que se pueden conducir a distancia, su principal característica es la precisión de su posicionamiento. Sus usos son más que un simple juguete ya que se les está encontrando un número de aplicaciones de gran utilidad, por ejemplo, en 2018 en la Feria de Electrónica de Consumo (Consume Electronics Show) en las Vegas presentaron un modelo para la cabina de los futuros drones con pasajeros, de mano de la compañía Bell Helicopter, la empresa se dedica al diseño de drones con cámara para uso profesional pero dio salto en escala para crear un drone lo suficientemente potente como para transportar uno o dos pasajeros [1]. La forma de pilotar del Ar.Drone 2.0 es intuitiva y de fácil acceso ya que mediante un smartphone o tablet que contengan un sistema operativo iOS o Android permite conectarse a la propia red Wi-Fi que genera el cuadricóptero. Contiene un puerto USB en el que se puede conectar una memoria para almacenar las grabaciones de video y las captura de imágenes [2]. Mediante el uso del lenguaje gráfico LabVIEW se ejecutó la manipulación y control del drone con el uso de un paquete de compatibilidad llamado toolkit, el cual nos brinda la comunicación entre el

drone, la computadora y el acceso a la cámara [3]. La forma de pilotar de la mayoría de los drones es por medio de controles diseñados y programados en los que se puede manipular sus seis grados de libertad y su movimiento en Roll, Pitch, Vertical Speed y Yaw Speed. En este trabajo se presenta un diseño de pilotar diferente en el que el objetivo es no tener un control remoto si no que el usuario automatice rutinas mediante máquinas de estados en las que puede controlar sus movimientos, tiempo de vuelo y las direcciones en que se quiera que se almacenen las fotos y los videos, esto visualizándolo por medio de un entorno gráfico en LabVIEW con el que será fácil de interactuar y controlar, logrando así plantear sus necesidad y ejecutarlas.

II. METODOLOGIA

AR.Drone 2.0, es la segunda versión de este dispositivo en la cuales destaca que es un cuadricóptero que cuenta con cámaras para poder tomar fotos y videos, autonomía de funcionamiento de 18 minutos, alcance de hasta 50 metros en conexión inalámbrica y seis grados de libertad en los que realiza movimientos tales como Roll, Pitch, Yaw Speed y Vertical Speed, que le permiten poder desplazarse en cualquier entorno [4]. A continuación en la siguiente Fig. 1 se muestran los diferentes movimientos que realiza en sus seis grados de libertad.

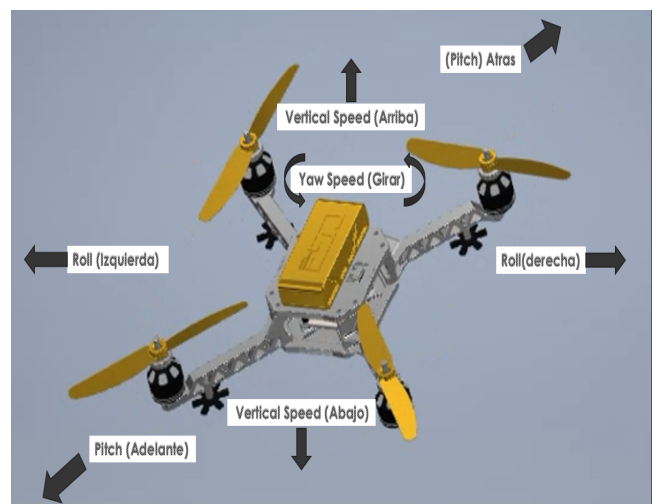


Fig.1 Movimientos de Roll, Pitch, Yaw Speed y Vertical Speed del Ar.Drone 2.0 en sus seis grados de libertad.

LabVIEW ofrece un enfoque de programación gráfica que ayuda a visualizar cada aspecto de la aplicación, incluyendo configuración de hardware, datos de medidas y depuración. Esta visualización hace que sea más fácil integrar hardware de medidas de cualquier proveedor, representar una lógica compleja en el diagrama, desarrollar algoritmos de análisis de datos, diseñar interfaces de usuario personalizadas y desarrollo de toolkits de compatibilidad como lo es el LVH AR Drone que ayuda a los aficionados y desarrolladores de LabVIEW y AR Drone a conectarse fácilmente con el Parrot compatible con las versiones Ar Drone 1.0 y 2.0. Este juego de herramientas ofrece una implementación nativa de LabVIEW del API del drone. Con ella, los programadores pueden crear aplicaciones que controlan completamente al drone usando un pequeño conjunto de parámetros de entrada [5].

Para cada rutina programada utilizamos ya anteriormente mencionado las herramientas de LabVIEW. La base para estas rutinas son las máquinas de estados, las cuales nos permitieron poder dividir cada paso de la secuencia. La máquina de estados es un sistema que dependiendo de una entrada de datos sea booleana o numérica, opera con ella y devuelve una salida.

Dividimos las rutinas en 4 (A), B), C), D)) en cada una de ellas se describe la manera en que se programaron y ejecutaron con el drone. A continuación se presentan las diferentes rutinas programas que se realizaron para el Ar.Drone 2.0 utilizando las máquinas de estados.

A. ENCENDER, DESPEGAR, SOBREVOLAR POR UN DETERMINADO TIEMPO Y DESCENDER.

Como primera rutina se programa que el drone encendiera, despegara, sobrevolara por un determinado tiempo y descendiera, esto con base a ciclos de recursividad, uso de comparadores y tiempos determinados por ejemplo de 5 min, 10 min, 15 min, al momento de que se cumplía el tiempo establecido de sobrevolar el drone desciende.

B. PROGRAMACIÓN DE SUS CUATRO TIPOS DE MOVIMIENTOS: ROLL, PITCH, VERTICAL SPEED Y YAW SPEED.

Como segunda tarea se programó que se moviera en sus cuatro formas: Yaw Speed, Roll, Pitch y Vertical Speed, para que el drone pudiera realizarlo utilizamos ciclos de recursividad, tiempos específicos en la máquina de estados. La máquina de estado nos permitió poder dividir los pasos de cada rutina en la que se define que se eleve, gire, se mueva a la derecha (Pitch) e izquierda (Pitch), los tiempos nos permiten poder definir la duración de cada estado y el ciclo de recursividad para que pasara de un estado a otro, en cada estado está definido el tipo de rutina que hace el drone. Para que el drone gire se definieron pasos cortos de 0.5 ya que si se insertan valores mayores se realizaran acciones más rápidas. Para descender definimos un condicional en el que si llega a un cierto estado

cumpliendo los anteriores salga del ciclo y descienda, terminando así su rutina.

C. TOMA DE FOTO.

Como tercera tarea se programó que el drone tomara fotos de un área, esto se realizó utilizando ciclos recursivos, comparadores, tiempos específicos y máquinas de estados. Se utilizó la librería para poder acceder a la cámara del drone, usando funciones como GetVideo Stream que es donde se inicializa el video. Decode Video es donde se convierte el arreglo bidimensional a formato de pixeles para poder ver la imagen. Para que las imágenes se guardaran y se pudieran visualizar definimos una dirección de locación diferente con extensión ya sea png o jpg en labview por ejemplo C:\Users\MAX\Desktop\Tesis\VISION_LABVIEW\Imag_Drone\prueba1.jpg en la que se guardara en el explorador de archivos ya sea en documentos, escritorio, descargas o memorias USB etc.

D. TOMA DE VIDEO.

Como cuarta rutina se programó que el drone pudiera tomar video y guárdalos en el explorador de archivos, para esto utilizamos ciclos recursivos, las herramientas para poder acceder a la cámara del drone GetVideo Stream y Decode Video. Posteriormente para poder grabar se utilizó primeramente el VI IMAQ AVI2 Get Codec Names que nos permite encontrar los códecs de compresión AVI. Cada códec tiene una calidad, tamaño y tiempo de compresión diferentes. Este VI solo enumera códecs compatibles con Video Compresión Manager (VCM). El VI, IMAQ AVI2 crea un nuevo archivo AVI o reescribe un archivo AVI antiguo en el que se definió una dirección en el explorador de archivos al igual que las imágenes pero con extensión .avi por ejemplo C:\Users\MAX\Desktop\Tesis\VISION_LABVIEW\video_drone\video5.avi. para observar el video tomado por el drone. Posteriormente utilizamos el VI AVI2 Write Frame VI este bloque escribe una imagen en el archivo AVI especificado por AVI Refnum. Se usó el IMAQ AVI2 Get Codec Names VI para encontrar los códecs de compresión AVI.

Para obtener una señal de video continua, realizamos modificaciones en el bloque al acceso de transmisión de la cámara GetVideo Stream del Ar.Drone 2.0, con la modificación obtuvimos una transición continua que nos permitió poder obtener por medio del drone un video observable y fotos visibles para el usuario. Cada uno de los resultados obtenidos fue en tiempo real y puestos en ejecución en áreas específicas utilizando el Ar.Drone 2.0 y la máquina de estados. En las máquinas de estados se establecieron los tiempos en los que se toma la foto, el video, el tipo de movimiento y el tiempo de vuelo. A continuación se presentan los resultados obtenidos en las siguientes Fig. 2, Fig. 3, y Fig. 4.

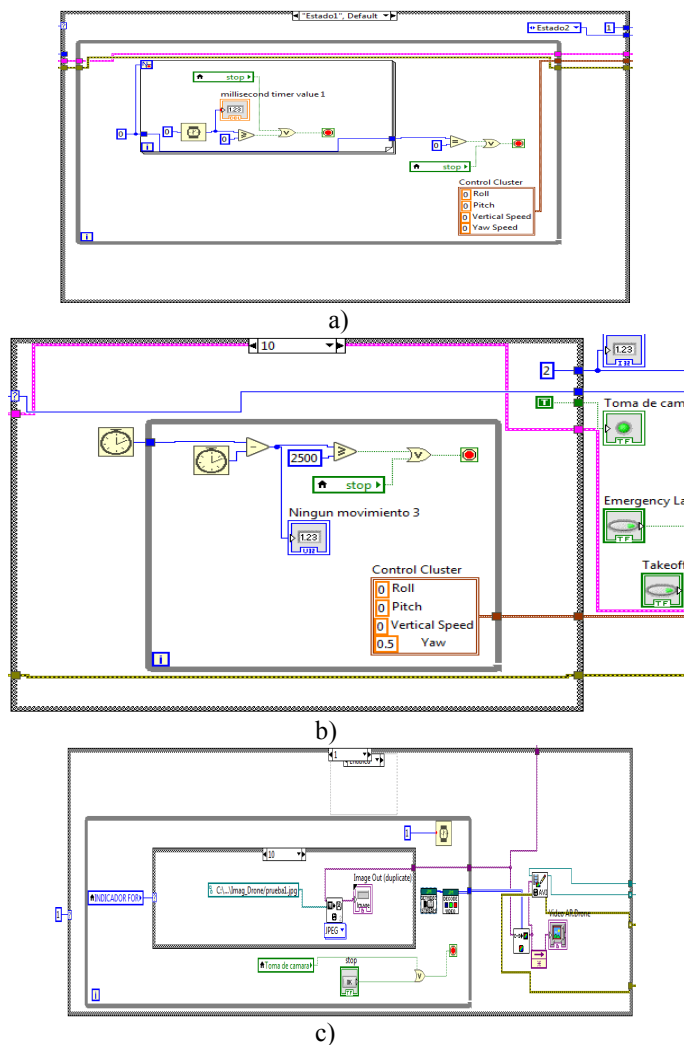


Fig.2 Uso de máquinas de estados en a) movimiento de grados de libertad. b) vuelo de tiempos específicos y c) toma de video.

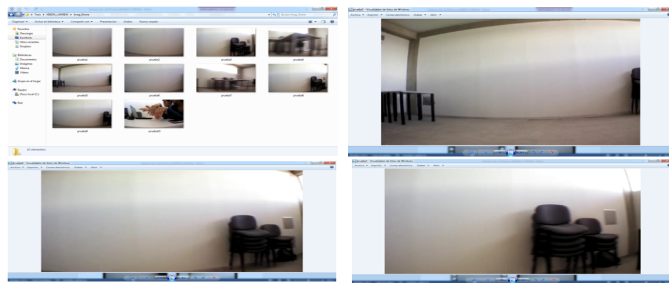


Fig.3 Toma de fotos de un área ejecutadas en LabVIEW y capturados por el Ar.Drone 2.0.

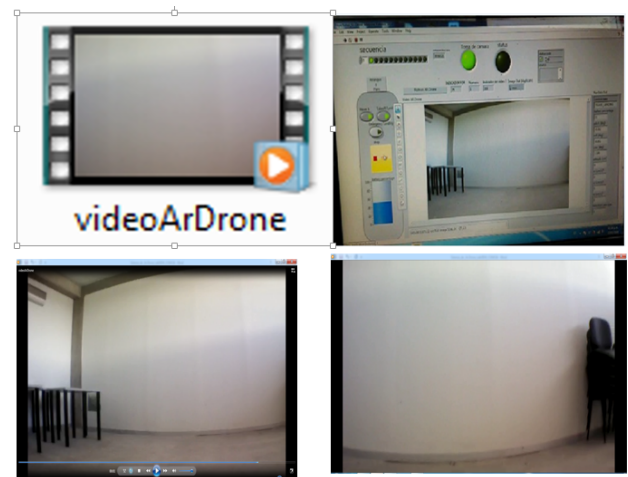


Fig.4 Toma de video de un área ejecutadas en LabVIEW y capturados por el Ar.Drone 2.0.

Los drones pueden tener un sinfín de aplicaciones ya sea en la investigación, búsqueda, adquisición de imágenes entre otros, superando cualquier tipo de obstáculos de mayor o menor impacto, y teniendo como ventaja el acceso a ellos ya que su costo es menor al de otros vehículos aéreos. En esta investigación se aplicó una nueva técnica para poder manipular a los drones Ar.Drone 2.0, con ésta cada rutina programada y puesta en ejecución arrojo resultados satisfactorios tanto en cada tipo de movimiento: Yaw Speed, Roll, Pitch y Vertical Speed, como la toma de imágenes y video que son visibles para el usuario teniendo como resultado una buena resolución de la cámara. A futuro se pretende realizar nuevas tareas, entre ellas que puedan reconocer caracteres mediante la técnica Optical Character Recognition (OCR).

III. REFERENCIAS

- [1] Galán, A. Ma. (22 de Febrero de 2018). *Levantar la mano y pedir drones taxi para dos personas*. Obtenido de Blogthinkbig.com: <http://blogthinkbig.com/ehang-pedir-taxi-dron>
- [2] Méndez, Á. (8 de AGO de 2012). *Quesabesde*. Obtenido de Parrot AR.Drone 2.0: análisis: http://www.quesabesde.com/noticias/parrot-ardrone-2-analisis-fotos_8986.
- [3] INSTRUMENTS, N. (2017). *sine*. Obtenido de AR Drone Toolkit para LabVIEW - LVH: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/211837>.
- [4] Parrot. (2016). *Parrot*. Obtenido de Parrot AR.Drone 2.0 Elite Edition: <https://www.parrot.com/us/es/drones/parrot-ardrone-20-elite-edition#técnicos>.
- [5] Corporation, N. I. (2018). *¿Qué es labview?* Obtenido de <http://www.ni.com/es-mx/shop/labview.html>.

